SANtricity ES Storage Manager Failover Drivers User Guide

Version 10.77

May 2011

51329-00, Rev. A

Revision History

Version and Date	Description of Changes
51329-00, Rev. A May 2011	Initial release of the document.

This document contains proprietary information of LSI Corporation and Sun Microsystems, Inc. The information contained herein is not to be used by or disclosed to third parties without the express written permission of an officer of LSI or Sun.

It is the policy of LSI and Sun to improve products as new technology, components, software, and firmware become available. We reserve the right to make changes to any products herein at any time without notice. All features, functions, and operations described herein may not be marketed in all parts of the world. In some instances, photographs and figures are of equipment prototypes. Therefore, before using this document, consult your sales representative or account team for information that is applicable and current. WE DO NOT ASSUME ANY RESPONSIBILITY OR LIABILITY FOR THE USE OF ANY PRODUCTS DESCRIBED HEREIN EXCEPT AS EXPRESSLY AGREED TO IN WRITING BY LSI .

LSI products are not intended for use in life-support appliances, devices, or systems. Use of any LSI product in such applications without written consent of the appropriate LSI officer is prohibited.

LSI Corporate Headquarters	Email	Website
Milpitas, CA 800-372-2447	globalsupport@lsi.com	www.lsi.com

Document Number: 51329-00, Rev. A

Table of Contents

Chapter 1: Overview of Failover Drivers

This topic describes how to use the various failover drivers for the Windows operating system, the Linux operating system, and Solaris operating system with SANtricity ES Storage Manager Version 10.75.

Failover drivers provide redundant path management for storage devices and cables in the data path from the host bus adapter to the controller. For example, you can connect two host bus adapters in the system to the redundant controller pair in a storage array, with different buses for each controller. If one host bus adapter, one bus cable, or one controller fails, the failover driver automatically reroutes input/output (I/O) to the good path, which permits the storage array to continue operating without interruption.

Failover drivers provide these functions:

- They automatically identify redundant I/O paths.
- They automatically reroute I/O to an alternate controller when a controller fails or all of the data paths to a controller fail.
- They check the state of known paths to the storage array.
- They provide status information on the controller and the bus.
- They check to see if the Service mode is enabled and if the modes have switched between Redundant Dual Active Controller (RDAC) and Auto-Volume Transfer (AVT).

Supported Failover Drivers Matrix

Table 1 Matrix of Supported Failover Drivers by Operating System (OS)

	Windows OS	Red Hat Enterprise Linux (RHEL) 4 OS Update 8 and RHEL 5 OS Update 4	SUSE Linux Enterprise (SLES) 10 OS Service Pack 3 and SLES 11 OS	Solaris 9 OS and Solaris 10 OS
Failover driver type	MPIO	RDAC	RDAC	MPxIO
Storage array mode	Either Mode Select or AVT	Either Mode Select or AVT	Either Mode Select or AVT	Mode Select
Number of paths supported	4 (default), 32 maximum	4 (default), 32 maximum	4 (default), 32 maximum	4
Number of volumes supported	255	256 for the Linux 2.4 OS 256 – 1 for the Linux 2.6 OS	256 for the Linux 2.4 OS 256 – 1 for the Linux 2.6 OS	255
Failover through single host bus adapter (HBA) support?*	Yes, as long as at least one good path to each controller is detected	Yes, as long as at least one good path to each controller is detected	Yes, as long as at least one good path to each controller is detected	Yes
Cluster support?	Yes	Yes	Yes	Yes

* Using failover through a single HBA support is not recommended.

Failover Driver Setup Considerations

Most storage arrays contain two controllers that are set up as redundant controllers. If one controller fails, the other controller in the pair takes over the functions of the failed controller, and the storage array continues to process data. You can then replace the failed controller and resume normal operation. You do not need to shut down the storage array to perform this task.

The redundant controller feature is managed by the failover driver software, which controls data flow to the controller pairs independent of the operating system (OS). This software tracks the current status of the connections and can perform the switch-over without any changes in the OS.

Whether your storage arrays have the redundant controller feature depends on a number of items:

- Whether the hardware supports it. Refer to the hardware documentation for your storage arrays to determine whether the hardware supports redundant controllers.
- Whether your OS supports certain failover drivers. Refer to the installation and support guide for your OS to determine if your OS supports redundant controllers.
- How the storage arrays are connected. The storage array must have two controllers installed in a redundant configuration. Redundant controllers can be configured only as an active/active pair. In an active/active pair, you can have multiple paths from the hosts to the active controller, and you perform load balancing on all of the active paths. Each controller has specific volumes assigned to it automatically. If one of the active controllers fails, the software automatically switches its assigned volumes to the other active controller.

Chapter 2: Failover Configuration Diagrams

You can configure failover in several ways. Each configuration has its own advantages and disadvantages. This section describes these configurations:

- Single-host configuration
- Multi-host configuration

This section also describes how the storage management software supports redundant controllers.

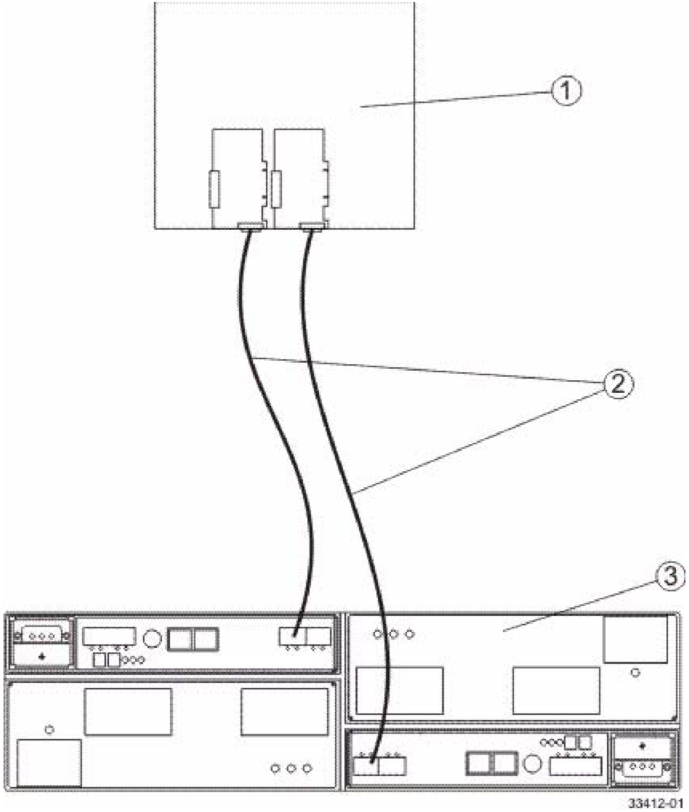
NOTE For best results, use the multi-host configuration. It provides the fullest failover protection and functionality in the event that a problem exists with the connection.

Single-Host Configuration

In a single-host configuration, the host system contains two host bus adapters (HBAs), with each HBA connected to one of the controllers in the storage array. The storage management software is installed on the host. The two connections are required for maximum failover support for redundant controllers.

Although you can have a single controller in a storage array or a host that has only one HBA port, you do not have complete failover data path protection with either of those configurations. The cable and the HBA become a single point of failure, and any data path failure could result in unpredictable effects on the host system. For the greatest level of I/O protection, provide each controller in a storage array with its own connection to a separate HBA in the host system.

Figure 1Single-Host-to-Storage Array Configuration



1. Host System with Two Fibre Channel Host Bus Adapters
2. Fibre Channel Connection – Fibre Channel Connection Might Contain One or More Switches
3. Storage Array with Two Fibre Channel Controllers

Multi-Host Configuration

For best results, use the multi-host configuration. It provides the best failover protection and functionality in the event that a problem exists with the connection.

In a multi-host configuration, two host systems are each connected by two connections to both of the controllers in a storage array. SANtricity ES Storage Manager, including failover driver support, is installed on each host.

Not every operating system supports this configuration. Consult the restrictions in the installation and support guide specific to your operating system for more information. Also, the host systems must be able to handle the multi-host configuration. Refer to the applicable hardware documentation.

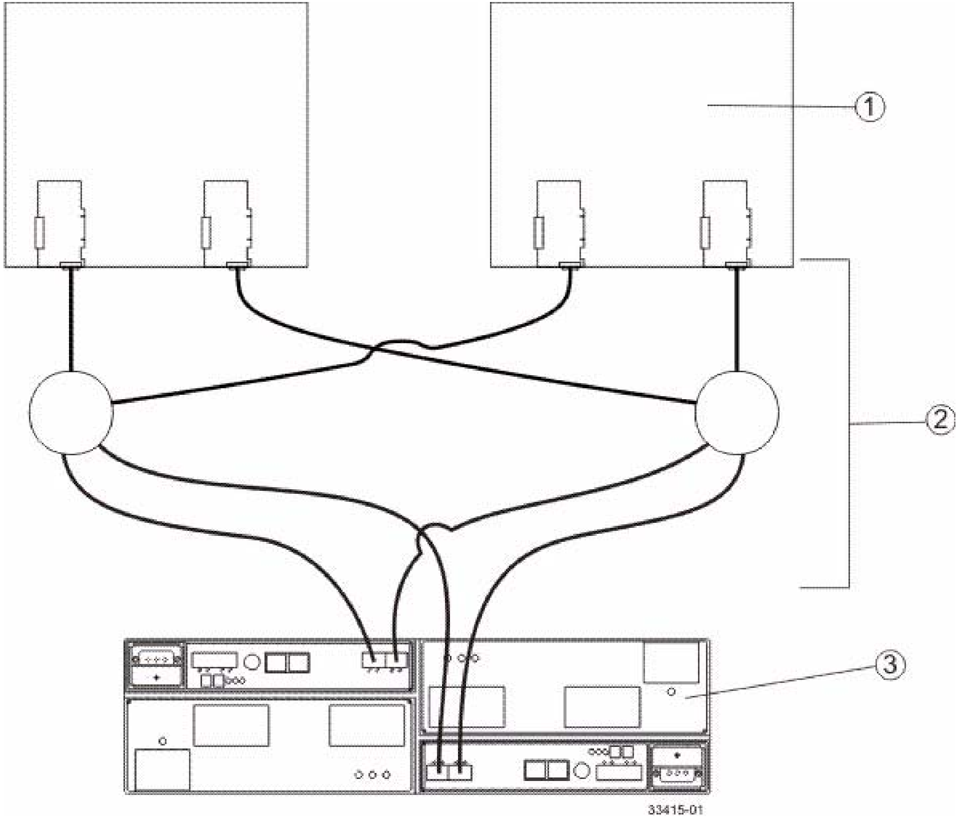
Both hosts have complete visibility of both controllers, all data connections, and all configured volumes in a storage array, plus failover support for the redundant controllers. However, in this configuration, you must use caution when you perform storage management tasks (especially deleting and creating volumes) to make sure that the two hosts do not send conflicting commands to the controllers in the storage arrays.

These items are unique to this configuration:

- Both hosts must have the same operating system version and SANtricity ES Storage Manager version installed.
- Both host systems must have the same volumes-per-host bus adapter capacity. This capacity is important for failover situations so that each controller can take over for the other and show all of the configured pools and volumes.

- If the operating system on the host system can create reservations, the storage management software honors them. This concept means that each host could have reservations to specified pools and volumes, and only the software on that host can perform operations on the reserved pool and volume. Without reservations, the software on either host system is able to start any operation. Therefore, you must use caution when you perform certain tasks that need exclusive access. Especially when you create and delete volumes, make sure that you have only one configuration session open at a time (from only one host), or the operations that you perform could fail.

Figure 2Multi-Host-to-Storage Array Configuration

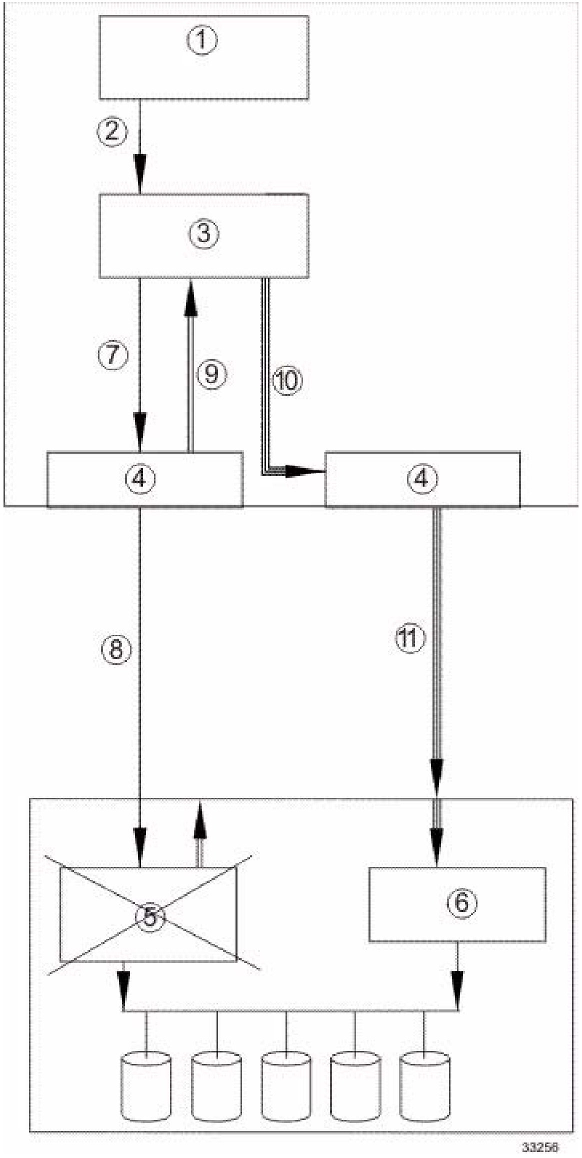


1. Two Host Systems, Each with Two Fibre Channel Host Bus Adapters
2. Fibre Channel Connections with Two Switches (Might Contain Different Switch Configurations)
3. Storage Array with Two Fibre Channel Controllers

Supporting Redundant Controllers

The following figure shows how failover drivers provide redundancy when the host application generates a request for I/O to controller A, but controller A fails. Use the numbered information to trace the I/O data path.

Figure 3Example of Failover I/O Data Path Redundancy



1. Host Application
2. I/O Request
3. Failover Driver
4. Host Bus Adapters
5. Controller A Failure
6. Controller B
7. Initial Request to the HBA
8. Initial Request to the Controller Failed
9. Request Returns to the Failover Driver
10. Failover Occurs and I/O Transfers to Another Controller
11. I/O Request Re-sent to Controller B

Chapter 3: How a Failover Driver Responds to a Data Path Failure

One of the primary functions of the failover feature is to provide path management. Failover drivers monitor the data path for devices that are not working correctly or for multiple link errors. If a failover driver detects either of these conditions, the driver automatically performs these steps:

- The failover driver checks the pair table for the redundant controller.
- The failover driver forces volumes to the other controller and routes all I/O to the remaining active controller.
- The older version of RDAC notifies you that an error has occurred with the Service Action Required LEDs on the storage array, and with a message that was sent to the error logs. The newer versions of RDAC and MPIO only send a message to the error logs.
- The failover driver performs a path failure if alternate paths to the same controller are available. If all of the paths to a controller fail, RDAC performs a controller failure.

A drive failure plus a controller failure are considered a double failure. The storage management software provides data integrity as long as all drive failures and controller failures are detected and fixed before more failures occur.

Chapter 4: Responding to a Data Path Failure

Use the Major Event Log (MEL) to respond to a data path failure. The information in the MEL provides the answers to these questions:

- What is the source of the error?
- What is required to fix the error, such as replacement parts or diagnostics?

The next step depends on whether you are a system administrator or a Sun Customer Care Center representative.

Responding to a Data Path Failure When You Are a System Administrator

Under most circumstances, contact your Sun Customer Care Center representative any time a path fails and the storage array notifies you of the failure. If your controller has failed and your storage array has customer-replaceable controllers, replace the failed controller. Follow the manufacturer's instructions for how to replace a failed controller.

Responding to a Data Path Failure When You Are a Sun Customer Care Center Representative

Use the Recovery Guru in the storage management software to diagnose and fix the problem, if possible. If you cannot fix the problem with the Recovery Guru, follow the manufacturer's instructions for how to replace a failed controller.

Chapter 5: Load-Balancing Policies

Load balancing is the redistribution of read/write requests to maximize throughput between the server and the storage array. Load balancing is very important in high workload settings or other settings where consistent service levels are critical. The multi-path driver transparently balances I/O workload without administrator intervention. Without multi-path software, a server sending I/O requests down several paths might operate with very heavy workloads on some paths, while other paths are not used efficiently.

The multi-path driver determines which paths to a device are in an active state and can be used for load balancing. The load-balancing policy uses one of three algorithms: round robin, least queue depth, or least path weight. Multiple options for setting the load-balancing policies let you optimize I/O performance when mixed host interfaces are configured. The load-balancing policies that you can choose depend on your operating system. Load balancing is performed on multiple paths to the same controller but not across both controllers.

Table 2 Load-Balancing Policies That Are Supported by the Operating Systems

Operating System	Multi-Path Driver	Load-Balancing Policy
Windows	MPIO DSM	Round robin with subset, least queue depth, weighted paths
Red Hat Enterprise Linux (RHEL)	RDAC	Round robin with subset, least queue depth
SUSE Linux Enterprise (SLES)	RDAC	Round robin with subset, least queue depth
Solaris	MPxIO	Round robin with subset

Least Queue Depth

The least queue depth policy is also known as the least I/Os policy or the least requests policy. This policy routes the next I/O request to the data path on the controller that owns the volume that has the least outstanding I/O requests queued. For this policy, an I/O request is simply a command in the queue. The type of command or the number of blocks that are associated with the command is not considered. The least queue depth policy treats large block requests and small block requests equally. The data path selected is one of the paths in the path group of the controller that owns the volume.

Round Robin with Subset I/O

The round robin with subset I/O load-balancing policy routes I/O requests, in rotation, to each available data path to the controller that owns the volumes. This policy treats all paths to the controller that owns the volume equally for I/O activity. Paths to the secondary controller are ignored until ownership changes. The basic assumption for the round robin with subset I/O policy is that the data paths are equal. With mixed host support, the data paths might have different bandwidths or different data transfer speeds.

Least Weighted Paths

The least weighted paths policy assigns a weight factor to each data path to a volume. An I/O request is routed to the path with the lowest weight value to the controller that owns the volume. If more than one data path to the volume has the same weight value, the round-robin with subset path selection policy is used to route I/O requests between the paths with the same weight value.

Chapter 6: Configuring Failover Drivers for the Windows OS and the Linux OS

NOTE This topic applies to both the Windows OS and the Linux OS.

Dividing I/O Activity Between Two RAID Controllers to Obtain the Best Performance

For the best performance of a redundant controller system, use the storage management software to divide I/O activity between the two RAID controllers in the storage array. You can use either the graphical user interface (GUI) or the command line interface (CLI).

To use the GUI to divide I/O activity between two RAID controllers, perform one of these steps:

- **Specify the owner of the preferred controller of an existing volume** – Select **Volume >> Change >> Ownership/Preferred Path** in the Array Management Window.

NOTE You can also use this method to change the preferred path and ownership of all volumes in a pool at the same time.

- **Specify the owner of the preferred controller of a volume when you are creating the volume** – Select **Volume >> Create** in the Array Management Window.

To use the CLI, go to the *Create RAID Volume (Free Extent Based Select)* online help topic for the command syntax and description.

Changing the Preferred Path Online Without Stopping the Applications

You can change the preferred path setting for a volume or a set of volumes online and without stopping the applications. If AVT is not enabled, the driver uses the new preferred path immediately. However, if AVT is enabled, the driver does not recognize that the preferred path has changed until the next cycle of the state change monitor. Therefore, the driver might continue to use the old preferred path for up to 60 seconds, or for the period to which the `ScanInterval` parameter is set. Because the driver continues to use the non-preferred path for a short period of time, the driver might trigger a volume not on preferred path Needs Attention condition in the storage management software. This condition is removed as soon as the state change monitor is run. A MEL event and an associated alert notification are delivered for the volume that is not on preferred path condition. If the driver needs some time to recognize that the preferred path has changed, you can configure the AVT alert delay period with the storage management software. The Needs Attention reporting is postponed until the driver failback task has had a chance to run.

NOTE The newer versions of the RDAC driver and the DSM driver do not recognize any AVT status change (enabled or disabled) until the next cycle of the state change monitor.

Chapter 7: Failover Drivers for the Windows Operating System

The failover driver for hosts with Microsoft Windows operating systems is Microsoft Multipath I/O (MPIO) with a Device Specific Module (DSM) for SANtricity ES Storage Manager.

Microsoft Multipath Input/Output

Microsoft Multipath I/O (MPIO) provides an infrastructure to build highly available solutions for Windows operating systems (OSs). MPIO uses Device Specific Modules (DSMs) to provide I/O routing decisions, error analysis, and failover.

NOTE You can use MPIO for all controllers that run controller firmware version 6.19 or later. MPIO is not supported for any earlier versions of the controller firmware, and MPIO cannot coexist on a server with RDAC. If you have legacy systems that run controller firmware versions earlier than 6.19, you must use RDAC for your failover driver. For SANtricity ES Storage Manager Version 10.10 and later and all versions of SANtricity Storage Manager, the Windows OS supports only MPIO.

Windows OS Restrictions

The MPIO DSM failover driver comes in these versions:

- 32-bit (x86)
- 64-bit Intel (Itanium or IA64)
- 64-bit AMD/EM64T (x64)

These versions are not compatible with each other. Because multiple copies of the driver cannot run on the same system, each subsequent release is backward compatible. In other words, a SANtricity ES Storage Manager Version 10.60 failover driver supports storage management software version 9.23.

You can use the DSM driver for all of the controllers that run controller firmware version 6.19 or later. The DSM driver is not supported for any earlier versions of the controller firmware, and it cannot coexist on a server with RDAC. If you have legacy systems that run controller firmware versions earlier than 6.19, you must use RDAC for your failover driver.

Native SCSI-2 Release/Reservation Commands in a Multipath Environment

If multiple paths exist to a single controller and a SCSI-2 release/reservation (R/R) is received for a volume, the DSM driver selects one path to each controller and repeats the request (called a reservation path). This function is necessary because the controllers cannot accept SCSI-2 R/R requests through multiple paths for a given volume. After the reservation path has been established, subsequent I/O requests for a volume are restricted to that path until a SCSI-2 release command is received. The DSM driver distributes the reservation paths if multiple volumes are mapped to the host, which distributes the load across multiple paths to the same controller.

Translating SCSI-2 Reservation/Release Commands to SCSI-3 Persistent Reservations

The DSM driver also supports the ability to translate the SCSI-2 R/R commands into SCSI-3 persistent reservations. This function allows a volume to use one of the previously mentioned load-balancing policies across all of the available controller paths rather than being restricted to a single reservation path. This feature requires the DSM driver to establish a unique “reservation key” for each host. This key is stored in the Registry and is named S2toS3Key. If this key is present, translations are performed, or else the “cloning” method is used.

Per-Protocol I/O Timeout Values

The timeout value associated with a non-passthrough I/O requests, such as read/write requests, is based on the MS driver's `TimeOutValue` parameter, as defined in the Registry. A feature within the DSM allows a customized timeout value to be applied based on the protocol, such as Fibre Channel, SAS, or iSCSI, that a path uses. Per-protocol timeout values provide these benefits:

- Without per-protocol timeout values, the `TimeOutValue` setting is global and affects all storage.
- The `TimeOutValue` is typically reset when an HBA driver is upgraded.
- For Windows Server 2003, the default disk timeout value may be adjusted based on the size of the I/O request. Adjusting the default disk timeout value helps support legacy SCSI devices.
- The DSM feature allows a more predictable timeout setting for Windows Server 2003 environments. For information about the configurable parameters associated with this feature, go to Configuration Settings for Windows DSM and Linux RDAC.

The per-protocol timeout values feature slightly modifies the way in which the `SynchTimeout` parameter is evaluated. The `SynchTimeout` parameter determines the I/O timeout for synchronous requests generated by the DSM driver. Examples include the SCSI-2 to SCSI-3 PR translations and inquiry commands used during device discovery. It is important that the timeout value for the requests from the DSM driver be at least as large as the per-protocol I/O timeout value. When a host boots, the DSM driver performs these actions:

- If the value of the `SynchTimeout` parameter is defined in the Registry key of the DSM driver, record the current value.
- If the value of the `TimeOutValue` parameter of the MS driver is defined in the Registry, record the current value.
- Use the higher of the two values as the initial value of the `SynchTimeout` parameter.
- If neither value is defined, use a default value of 10 seconds.
- For each synchronous I/O request, the higher value of either the per-protocol I/O timeout or the `SynchTimeout` parameter is used. For example:
  - — If the value of the `SynchTimeout` parameter is 120 seconds, and the value of the `TimeOutValue` parameter is 60 seconds, 120 seconds is used for the initial value.
  - — If the value of the `SynchTimeout` parameter is 120 seconds, and the value of the `TimeOutValue` parameter is 180 seconds, 180 seconds is used for the initial value of the synchronous I/O requests for the DSM driver.
  - — If the I/O timeout value for a different protocol (for example, SAS) is 60 seconds and the initial value is 120 seconds, the I/O will be sent using a 120-second timeout.

Selective LUN Transfer

This feature limits the conditions under which the DSM will move a LUN to the alternative controller to three cases:

1. When a DSM with a path to only one controller, the non-preferred path, discovers a path to the alternate controller.
2. When an I/O request is directed to a LUN that is owned by the preferred path, but the DSM is attached to only the non-preferred path.
3. When an I/O request is directed to a LUN that is owned by the non-preferred path, but the DSM is attached to only the preferred path.

Cases 2 and 3 have these user-configurable parameters that can be set to tune the behavior of this feature.

- The maximum number of times that the LUN transfer will be issued. This parameter setting prevents a continual ownership thrashing condition from occurring in cases where the controller module or the array module is attached to another host that requires the LUN be owned by the current controller.
- A time delay before LUN transfers are attempted. This parameter is used to de-bounce intermittent I/O path link errors. During the time delay, I/O requests will be retried on the current controller to take advantage of the possibility that another host might transition the LUN to the current controller.

For further information on these two parameters, go to Configuration Settings for Windows DSM and Linux RDAC.

In the case where the host system is connected to both controllers and an I/O is returned with a 94/01 status (the LUN is not owned and can be owned), the DSM will modify its internal data on which controller to use for that LUN and reissue the command to the other controller. The DSM will not issue a LUN transfer command to the controller module or array module to avoid interfering with other hosts that might be attached to that controller module or the array module.

When the DSM detects that a volume-transfer operation is required, the DSM will not immediately issue the command. It will delay for three seconds before sending the command to the controller module or the array module. This delay is to attempt to batch together as many volume-transfer operations for other LUNs as possible. This batching method is used because the controller single-threads volume transfer operations and will reject additional transfer commands until the controller has completed the operation it is currently working on. This single-threading behavior extends the period of time that I/Os are not being successfully serviced by the controller module or the array module.

This feature will be enabled if these conditions exist:

- The controller module or the array module does not have AVT enabled.
- The DSM configurable parameter `ClassicModeFailover` is set to `1`.
- The DSM configurable parameter `DisableLunRebalance` is set to 4.

Windows Failover Cluster

Clustering for the Windows Server 2008 OS and the Windows Server 2008 R2 OS uses SCSI-3 persistent reservations natively. As a result, the DSM driver does not perform translations for any SCSI-2 R/R commands, and you can use one of the previously mentioned load-balancing policies across all controller paths. Translations still occur if the DSM driver is running in a Windows Server 2003 OS-based environment. When using clustering, set the `DisableLunRebalance` parameter to 3. For information about this parameter, go to Configuration Settings for Windows DSM and Linux RDAC.

Reduced Failover Timing

Settings related to drive I/O timeout and HBA connection loss timeout are adjusted in the host operating system so that failover does not occur when a controller is restarted. These settings provide protection from exception conditions that might occur when both controllers in a controller module or an array module are restarted at the same time, but they have the unfortunate side-effect of causing longer failover times than may be tolerated by some application or clustered environments. Support for the reduced failover timing feature includes support for reduced timeout settings, which result in faster failover response times.

The following restrictions apply to this feature:

- Only the Windows Server 2008 OS and the Windows Server 2008 R2 OS support this feature.
- Non-enterprise products attached to a host must use controller firmware release 7.35 or higher. Enterprise products attached to a host must use controller firmware release 7.6 or higher. For configurations where a mix of earlier releases is installed, older versions are not supported.
- When this feature is used with Windows Server Failover Cluster (WSFC) on the Windows Server 2008 OS, MPIO HotFix 970525 is required. The required HotFix is a standard feature for the Windows Server 2008 R2 OS.

Additional restrictions apply to storage array brownout conditions. Depending on how long the brownout condition lasts, PR registration information for volumes might be lost. By design, WSFC periodically polls the cluster storage to determine the overall health and availability of the resources. One action performed during this polling is a PRIN READ_KEYS request, which returns registration information. Because a brownout condition can cause blank information to be returned, WSFC interprets this as a loss of access to the drive resource and attempts recovery by first failing that drive resource, and then performing a new arbitration.

NOTE Any condition that causes blank registration information to be returned, where previous requests returned valid registration information, can cause the drive resource to fail. If the arbitration succeeds, the resource is brought online. Otherwise, the resource remains in a failed state. One reason for an arbitration failure is the combination of brownout condition and PnP timing issues if the HBA timeout period expires. When the timeout period expires, the OS is notified of an HBA change and must re-enumerate the HBAs to determine which devices no longer exist or, in the case where a connection is re-established, what devices are now present.

The arbitration recovery process happens almost immediately after the resource is failed. This situation, along with the PnP timing issue, can result in a failed recovery attempt. Fortunately, you can modify the timing of the recovery process by using the `cluster. exe` command-line tool. Microsoft recommends changing the following, where *resource_name* is a cluster disk resource, such as Cluster Disk 1:

```
cluster.exe resource “resource_name” /prop RestartDelay=4000
cluster.exe resource “resource_name” /prop RestartThreshold=5
```

The previous example changes the disk-online delay to four seconds and the number of online restarts to five. The changes must be repeated for each drive resource. The changes will persist across reboots. To display the current (or changed) settings, use the following command:

```
cluster.exe resource “resource_name” /prop
```

Another option exists to prevent the storage array from returning blank registration information. This option takes advantage of the Active Persist Through Power Loss (APTPL) feature found in Persistent Reservations, which ensures that the registration information persists through brownout or other conditions related to a power failure. A PTPL is enabled when a registration is initially made to the drive resource. WSFC does not use the APTPL feature, but an option is provided in the LSI DSM to set this feature when a registration request is made.

NOTE Because the APTPL feature is not supported in WSFC, Microsoft does not recommend its use. The APTPL feature should be considered as an option of last resort when the `cluster. exe` options cannot meet the tolerances needed. If a cluster setup cannot be brought online successfully after this option is used, the controller shell or SYMbol commands might be required to clear existing persistent reservations.

NOTE The APTPL feature within the DSM is enabled using the DSM utility with the `- o` (feature) option by setting the `SetAPTPLForPR` option to `1`. According to the SCSI specification, you must this option before PR registration occurs. If you set this option after a PR registration occurs, take the disk resource offline, and then bring the disk resource back online. If the DSM has set the APTPL option during registration, an internal flag is set, and the DSM utility output from the `-g` option indicates this condition. The SCSI specification does not provide a means for the initiator to query the storage array to determine the current APTPL setting. As a result, the `-g` output from one node might show the option set, but another node might not. Interpret output from the `-g` option with caution. By default, the DSM is released without this option enabled.

Wait Time Settings

When the failover driver receives an I/O request for the first time, the failover driver logs timestamp information for the request. If a request returns an error and the failover driver decides to retry the request, the current time is compared with the original timestamp information. Depending on the error and the amount of time that has elapsed, the request is retried to the current owning controller for the LUN, or a failover is performed and the request sent to the alternate controller. This process is known as a wait time. If the `NotReadyWaitTime` value, the `BusyWaitTime` value, and the `QuiescenceWaitTime` value are greater than the `ControllerIoWaitTime` value, they will have no effect.

For the Linux OS, the configuration settings can be found in the `/etc/mpp.conf` file. For the Windows OS, the configuration settings can be found in the Registry under:

```
HKEY_LOCAL_MACHINE\System\CurrentControlSet\ Services\<DSM_Driver>
```

In the preceding setting, *`<DSM_Driver>`* is the name of the OEM-specific driver. The default driver is named `mppdsm.sys`. Any changes to the settings take effect the next time the host is restarted.

ATTENTION Possible loss of data access – If you change these settings from their configured values, you might lose access to the storage array.

Table 3 Configuration Settings for the Path Congestion Detection Feature

Parameter Name	Default Value (Operating System)	Description
`NotReadyWaitTime`	`300` (Windows) `270` (Linux)	The time, in seconds, a Not Ready condition (SK 0x06, ASC/ASCQ 0x04/0x01) is allowed before failover is performed. Valid values range from `0x1` to `0xFFFFFFFF`.
`BusyWaitTime`	`600` (Windows) `270` (Linux)	The time, in seconds, a Busy condition is allowed before a failover is performed. Valid values range from `0x1` to `0xFFFFFFFF`.

Parameter Name	Default Value (Operating System)	Description
QuiescenceWaitTime	600 (Windows) 270 (Linux)	The time, in seconds, a Busy condition is allowed before a failover is performed. Valid values range from 0x1 to 0xFFFFFFFF.
ControllerIoWaitTime	600 (Windows) 120 (Linux)	Provides an upper-bound limit, in seconds, that an I/O is retried on a controller regardless of retry status before a failover is performed. If the limit is exceeded on the alternate controller, the I/O is again attempted on the original controller. This process continues until the value of the ArrayIoWaitTime limit is reached. Valid values range from 0x1 to 0xFFFFFFFF.
ArrayIoWaitTime	600 (Windows DSM) 600 (Linux RDAC)	Provides an upper-bound limit, in seconds, that an I/O is retried to the storage array regardless of to which controller the request is attempted. After this limit is exceeded, the I/O is returned with a failure status. Valid values range from 0x1 to 0xFFFFFFFF.

Path Congestion Detection and Online/Offline Path States

The path congestion detection feature allows the DSM driver to place a path offline based on the path I/O latency. The DSM will automatically set a path offline when I/O response times exceed user-definable congestion criteria. An administrator can manually place a path into the Admin Offline state. When a path is either set offline by the DSM or by an administrator, I/O will be routed to a different path. The offline or admin offline path will not be used for I/O until the system administrator sets the path online.

For more information on path congestion configurable parameters, go to Configuration Settings for Windows DSM and Linux RDAC.

Configuration Settings for Windows DSM and Linux RDAC

This topic applies to both the Windows OS and the Linux OS. The failover driver that is provided with the storage management software contains configuration settings that can modify the behavior of the driver.

- For the Linux OS, the configuration settings are in the `/etc/mpp.conf` file.
- For the Windows OS, the configuration settings are in the `HKEY_LOCAL_MACHINE\System\CurrentControlSet\ Services\<DSM_Driver>\Parameters` registry key, where <*DSM_Driver*> is the name of the OEM-specific driver.

The default driver is `mppdsm.sys`. Any changes to the settings take effect on the next reboot of the host.

The default values listed in the following table apply to both the Windows OS and the Linux OS unless the OS is specified in parentheses. Many of these values are overridden by the failover installer for the Linux OS or the Windows OS.

ATTENTION Possible loss of data access – If you change these settings from their configured values, you might lose access to the storage array.

Table 4 Configuration Settings for Windows DSM and Linux RDAC

Parameter Name	Default Value (Operating System)	Description
`MaxPathsPerController`	4	The maximum number of paths (logical endpoints) that are supported per controller. The total number of paths to the storage array is the `MaxPathsPerController` value multiplied by the number of controllers. The allowed values range from `0x1` (1) to `0x20` (32) for Windows, and from `0x1` (1) to `0xFF` (255) for Linux RDAC. For use by Sun Customer Care Center representatives only.
`ScanInterval`	`1` (Windows) `60` (Linux)	The interval time, in seconds, that the failover driver will check for these conditions: ■ A change in preferred ownership for a LUN ■ An attempt to rebalance LUNs to their preferred paths ■ A change in AVT enabled status or disabled status For the Windows OSs, the allowed values range from `0x1` to `0xFFFFFFFF` and must be specified in minutes. For the Linux OSs, the allowed values range from `0x1` to `0xFFFFFFFF` and must be specified in seconds. For use by Sun Customer Care Center representatives only.
`ErrorLevel`	3	This setting determines which errors to log. These values are valid: ■ `0` – Display all errors ■ `1` – Display path failover errors, controller failover errors, retryable errors, fatal errors, and recovered errors ■ `2` – Display path failover errors, controller failover errors, retryable errors, and fatal errors ■ `3` – Display path failover errors, controller failover errors, and fatal errors ■ `4` – Display controller failover errors, and fatal errors For use by Sun Customer Care Center representatives only.
`SelectionTimeoutRetryCount`	`0`	The number of times a selection timeout is retried for an I/O request before the path fails. If another path to the same controller exists, the I/O is retried. If no other path to the same controller exists, a failover takes place. If no valid paths exist to the alternate controller, the I/O is failed. The allowed values range from `0x0` to `0xFFFFFFFF`. For use by Sun Customer Care Center representatives only.

Parameter Name	Default Value (Operating System)	Description
`CommandTimeoutRetryCount`	`1`	The number of times a command timeout is retried for an I/O request before the path fails. If another path to the same controller exists, the I/O is retried. If another path to the same controller does not exist, a failover takes place. If no valid paths exist to the alternate controller, the I/O is failed. The allowed values range from `0x0` to `0xa` (10) for Windows, and from `0x0` to `0xFFFFFFFF` for Linux RDAC. For use by Sun Customer Care Center representatives only.
`UaRetryCount`	`10`	The number of times a Unit Attention (UA) status from a LUN is retried. This parameter does not apply to UA conditions due to Quiescence In Progress. The allowed values range from `0x0` to `0x64` (100) for Windows, and from `0x0` to `0xFFFFFFFF` for Linux RDAC. For use by Sun Customer Care Center representatives only.
`SynchTimeout`	`120`	The timeout, in seconds, for synchronous I/O requests that are generated internally by the failover driver. Examples of internal requests include those related to rebalancing, path validation, and issuing of failover commands. The allowed values range from `0x1` to `0xFFFFFFFF`. For use by Sun Customer Care Center representatives only.
`DisableLunRebalance`	`0`	This parameter provides control over the LUN failback behavior of rebalancing LUNs to their preferred paths. These values are possible: ■ `0` – LUN rebalance is enabled for both AVT and non-AVT modes. ■ `1` – LUN rebalance is disabled for AVT mode and enabled for non-AVT mode. ■ `2` – LUN rebalance is enabled for AVT mode and disabled for non-AVT mode. ■ `3` – LUN rebalance is disabled for both AVT and non-AVT modes. ■ `4` – The selective LUN Transfer feature is enabled if AVT mode is off and `ClassicModeFailover` is set to LUN level 1.
`S2ToS3Key`	Unique key	This value is the SCSI-3 reservation key generated during failover driver installation. **NOTE** For use by Sun Customer Care Center representatives only.

Parameter Name	Default Value (Operating System)	Description
`LoadBalancePolicy`	`1`	This parameter determines the load-balancing policy used by all volumes managed by the Windows DSM and Linux RDAC failover drivers. These values are valid: ■ `0` – round robin with subset. ■ `1` – Least queue depth with subset. ■ `2` – Least path weight with subset (Windows OS only).
`ClassicModeFailover`	`0`	This parameter provides control over how the DSM handles failover situations. These values are valid: ■ `0` – Perform controller-level failover (all LUNs are moved to the alternate controller). ■ `1` – Perform LUN-level failover (only the LUNs indicating errors are transferred to the alternate controller).
`SelectiveTransferMaxTransferAttempts`	`3`	This parameter sets the maximum number of times that a host will transfer the ownership of a LUN to the alternate controller when the Selective LUN Transfer mode is enabled. This setting prevents multiple hosts from continually transferring LUNs between controllers.
`SelectiveTransferMinIOWaitTime`	`5`	This parameter sets the minimum wait time (in seconds) that the DSM will wait before transferring a LUN to the alternate controller when the Selective LUN Transfer mode is enabled. This parameter tries to stop excessive LUN transfers due to intermittent link errors.

Example Configuration Settings for the Path Congestion Detection Feature

NOTE Before path congestion detection can be enabled, you must set the `CongestionResponseTime`, `CongestionTimeFrame`, and `CongestionSamplingInterval` parameters to valid values.

To set the path congestion IO response time to 10 seconds:

```
dsmUtil -o CongestionResponseTime=10,SaveSettings
```

To set the path congestion sampling interval to one minute:

```
dsmUtil -o CongestionSamplingInterval=60
```

To enable path congestion detection:

```
dsmUtil -o CongestionDetectionEnabled=0x1,SaveSettings
```

To use the `dsmUtil -o` command to set a path to Admin Offline:

```
dsmUtil -o SetPathOffline=0x77070001
```

NOTE The path ID (in this example `0x77070001`) is found using the `dsmUtil -g` command.

To use the `dsmUtil -o` command to set a path to online:

```
dsmUtil -o SetPathOnline=0x77070001
```

Device Specific Module for the Microsoft MPIO Solution

The DSM driver is the hardware-specific part of Microsoft's MPIO solution. This release supports Microsoft's Windows Server 2003 OS, Windows Server 2008 OS, and Windows Server 2008 R2 OS. The Hyper-V role is also supported when running the DSM within the parent partition. The DSM provides these features for SANtricity ES Storage Manager Version 10.75.

The directory structures concerning the DSM driver include these paths:

- `\Device\MPPDSM` – This structure contains information that is maintained by the DSM driver.
- `\Device\Scsi` – This structure contains information that is maintained by the ScsiPort driver.

The name MPPDSM might be different if a non-LSI generic solution is installed.

Table 5 Object Path and Descriptions of the WinObj DSM

Object Path	Description
`\Device\MPPDSM`	The root directory for all named objects that are created by the DSM driver.
`\Device\MPPDSM\<storage array>`	The root directory for all named objects that are created by the storage array named `<storage array>`.
`\Device\MPPDSM\<storage array>\<ctlr>`	The root directory for all named objects that are created for a given controller. The `<ctlr>` value can either be A or B.
`\Device\MPPDSM\<storage array>\<ctlr>\P<port>P<path>I<id>`	The root directory for all named objects that are created for a given path to a controller. The `<port>` value, the `<path>` value, and the `<id>` value represent a SCSI address from a given HBA port.
`\Device\Scsi`	The root directory for all of the named objects created by the ScsiPort driver. Each object represents a physical path found by a given HBA.
`\Device\Scsi\<adapter>Port<port>\Path<path>Target<target>Lun<lun>`	ScsiPort-based HBA drivers. A named device object that represents a drive. The `<adapter>` value represents the HBA vendor. For QLogic, this value is based on the HBA model number (for example, ql2300). The `<port>` value, the `<path>` value, and the `<target>` value represent the location of the volume on the HBA.
`\Device\<auto-generated id>`	StorPort-based HBA drivers. An auto-generated named device object representing a drive.

With this information, you can reach these conclusions:

- The objects shown in the `\Device\Scsi` directory show the physical volumes that are identified by the HBAs. If a specific volume is not in this list, the DSM driver cannot detect the volumes.
- The objects shown in the `\Device\MPPDSM` directory show the items that are reported by MPIO to the DSM driver. If a device is not in this list, MPIO has not notified the DSM driver.

Device Specific Module Driver Directory Structures

NOTE The name MPPDSM in the directory structures might be different based on your network configuration.

The directory structures for the DSM driver include these paths:

- `\Device\MPPDSM` – This structure contains information that is maintained by the DSM driver. The objects shown in the `\Device\MPPDSM` directory in the following table show the items that are reported by MPIO to the DSM driver. If a device is not in this list, MPIO has not notified the DSM driver.

- `\Device\Scsi` – This structure contains information that is maintained by the ScsiPort driver. The objects shown in the `\Device\Scsi` directory in the following table show the physical volumes that are identified by the HBAs. If a specific volume is not in this list, the DSM driver cannot detect the volumes.

Table 6 Object Path and Descriptions of the WinObj DSM

Object Path	Description
`Device\MPPDSM`	The root directory for all named objects that are created by the DSM driver.
`\Device\MPPDSM\<`*`storage array`*`>`	The root directory for all named objects that are created by the storage array named <*storage array*>.
`\Device\MPPDSM\<`*`storage array`*`>`	The root directory for all named objects that are created by the storage array named <*storage array*>.
`\Device\MPPDSM\<`*`storage array`*`>\<`*`ctlr`*`>`	The root directory for all named objects that are created for a given controller. The <*ctlr*> value can either be A or B.
`\Device\MPPDSM\<`*`storage array`*`>\<`*`ctlr`*`>\P<`*`port`*`>P<`*`path`*`>I<`*`id`*`>`	The root directory for all named objects that are created for a given path to a controller. The <*port*> value, the <*path*> value, and the <*id*> value represent a SCSI address from a given HBA port.
`\Device\MPPDSM\<storage array>\<`*`ctlr`*`>\P<`*`port`*`>P<`*`path`*`>I<`*`id`*`>\<`*`lun`*`>`	The <*lun*> value represents the volume number assigned to the device for a given controller/path combination.
`\Device\Scsi`	The root directory for all of the named objects created by the ScsiPort driver. Each object represents a physical path found by a given HBA.
`\Device\Scsi\<`*`adapter`*`>Port<`*`port`*`>\Path<`*`path`*`>Target<`*`target`*`>Lun<`*`lun`*`>`	ScsiPort-based HBA drivers. A named device object that represents a drive. The <*adapter*> value represents the HBA vendor. For QLogic, this value is based on the HBA model number (for example, ql2300). The <*port*> value, the <*path*> value, and the <*target*> value represent the location of the volume on the HBA.
`\Device\<`*`auto-generated id`*`>`	StorPort-based HBA drivers. An auto-generated named device object representing a drive.

dsmUtil Utility

The dsmUtil utility is a command-line driven utility that works only with the Multipath I/O (MPIO) Device Specific Module (DSM) solution. The utility is used primarily as to tell the DSM driver to perform various maintenance tasks, but the utility can also serve as a troubleshooting tool when necessary.

Configuration Settings for Windows DSM and Linux RDAC

This topic applies to both the Windows OS and the Linux OS. The failover driver that is provided with the storage management software contains configuration settings that can modify the behavior of the driver.

- For the Linux OS, the configuration settings are in the `/etc/mpp.conf` file.
- For the Windows OS, the configuration settings are in the `HKEY_LOCAL_MACHINE\System\CurrentControlSet\ Services\<`*`DSM_Driver`*`>\Parameters` registry key, where <*DSM_Driver*> is the name of the OEM-specific driver.

The default driver is `mppdsm.sys`. Any changes to the settings take effect on the next reboot of the host.

The default values listed in the following table apply to both the Windows OS and the Linux OS unless the OS is specified in parentheses. Many of these values are overridden by the failover installer for the Linux OS or the Windows OS.

ATTENTION **Possible loss of data access** – If you change these settings from their configured values, you might lose access to the storage array.

Table 7 Configuration Settings for Windows DSM and Linux RDAC

Parameter Name	Default Value (Operating System)	Description
`MaxPathsPerController`	`4`	The maximum number of paths (logical endpoints) that are supported per controller. The total number of paths to the storage array is the `MaxPathsPerController` value multiplied by the number of controllers. The allowed values range from `0x1` (1) to `0x20` (32) for Windows, and from `0x1` (1) to `0xFF` (255) for Linux RDAC. For use by Sun Customer Care Center representatives only.
`ScanInterval`	`1` (Windows) `60` (Linux)	The interval time, in seconds, that the failover driver will check for these conditions: ■ A change in preferred ownership for a LUN ■ An attempt to rebalance LUNs to their preferred paths ■ A change in AVT enabled status or disabled status For the Windows OSs, the allowed values range from `0x1` to `0xFFFFFFFF` and must be specified in minutes. For the Linux OSs, the allowed values range from `0x1` to `0xFFFFFFFF` and must be specified in seconds. For use by Sun Customer Care Center representatives only.
`ErrorLevel`	`3`	This setting determines which errors to log. These values are valid: ■ `0` – Display all errors ■ `1` – Display path failover errors, controller failover errors, retryable errors, fatal errors, and recovered errors ■ `2` – Display path failover errors, controller failover errors, retryable errors, and fatal errors ■ `3` – Display path failover errors, controller failover errors, and fatal errors ■ `4` – Display controller failover errors, and fatal errors For use by Sun Customer Care Center representatives only.
`SelectionTimeoutRetryCount`	`0`	The number of times a selection timeout is retried for an I/O request before the path fails. If another path to the same controller exists, the I/O is retried. If no other path to the same controller exists, a failover takes place. If no valid paths exist to the alternate controller, the I/O is failed. The allowed values range from `0x0` to `0xFFFFFFFF`. For use by Sun Customer Care Center representatives only.

Parameter Name	Default Value (Operating System)	Description
`CommandTimeoutRetryCount`	`1`	The number of times a command timeout is retried for an I/O request before the path fails. If another path to the same controller exists, the I/O is retried. If another path to the same controller does not exist, a failover takes place. If no valid paths exist to the alternate controller, the I/O is failed. The allowed values range from `0x0` to `0xa` (10) for Windows, and from `0x0` to `0xFFFFFFFF` for Linux RDAC. For use by Sun Customer Care Center representatives only.
`UaRetryCount`	`10`	The number of times a Unit Attention (UA) status from a LUN is retried. This parameter does not apply to UA conditions due to Quiescence In Progress. The allowed values range from `0x0` to `0x64` (100) for Windows, and from `0x0` to `0xFFFFFFFF` for Linux RDAC. For use by Sun Customer Care Center representatives only.
`SynchTimeout`	`120`	The timeout, in seconds, for synchronous I/O requests that are generated internally by the failover driver. Examples of internal requests include those related to rebalancing, path validation, and issuing of failover commands. The allowed values range from `0x1` to `0xFFFFFFFF`. For use by Sun Customer Care Center representatives only.
`DisableLunRebalance`	`0`	This parameter provides control over the LUN failback behavior of rebalancing LUNs to their preferred paths. These values are possible: ■ `0` – LUN rebalance is enabled for both AVT and non-AVT modes. ■ `1` – LUN rebalance is disabled for AVT mode and enabled for non-AVT mode. ■ `2` – LUN rebalance is enabled for AVT mode and disabled for non-AVT mode. ■ `3` – LUN rebalance is disabled for both AVT and non-AVT modes. ■ `4` – The selective LUN Transfer feature is enabled if AVT mode is off and `ClassicModeFailover` is set to LUN level 1.
`S2ToS3Key`	Unique key	This value is the SCSI-3 reservation key generated during failover driver installation. **NOTE** For use by Sun Customer Care Center representatives only.

Parameter Name	Default Value (Operating System)	Description
`LoadBalancePolicy`	1	This parameter determines the load-balancing policy used by all volumes managed by the Windows DSM and Linux RDAC failover drivers. These values are valid: ■ `0` – round robin with subset. ■ `1` – Least queue depth with subset. ■ `2` – Least path weight with subset (Windows OS only).
`ClassicModeFailover`	0	This parameter provides control over how the DSM handles failover situations. These values are valid: ■ `0` – Perform controller-level failover (all LUNs are moved to the alternate controller). ■ `1` – Perform LUN-level failover (only the LUNs indicating errors are transferred to the alternate controller).
`SelectiveTransferMaxTransferAttempts`	3	This parameter sets the maximum number of times that a host will transfer the ownership of a LUN to the alternate controller when the Selective LUN Transfer mode is enabled. This setting prevents multiple hosts from continually transferring LUNs between controllers.
`SelectiveTransferMinIOWaitTime`	5	This parameter sets the minimum wait time (in seconds) that the DSM will wait before transferring a LUN to the alternate controller when the Selective LUN Transfer mode is enabled. This parameter tries to stop excessive LUN transfers due to intermittent link errors.

Windows DSM Configuration Settings

The following configuration settings are applied using the utility `dsmUtil -o` option parameter. Go to dsmUtil Utility.

Table 8 Configuration Settings for the Path Congestion Detection Feature

Parameter Name	Default Value	Description
`CongestionDetectionEnabled`	`0x0`	A Boolean value that indicates whether the path congestion detection is enabled. If this parameter is not defined or is set to `0x0`, the value is false, the path congestion feature is disabled, and all of the other parameters are ignored. If set to `0x1`, the path congestion feature is enabled. Valid values are `0x0` or `0x1`.
`CongestionResponseTime`	`0x0`	If `CongestionIoCount` is `0x0` or not defined, this parameter represents an average response time in seconds allowed for an I/O request. If the value of the `CongestionIoCount` parameter is non-zero, then this parameter is the absolute time allowed for an I/O request. Valid values range from `0x1` to `0x10000` (approximately 18 hours).
`CongestionIoCount`	`0x0`	The number of I/O requests that have exceeded the value of the `CongestionResponseTime` parameter within the value of the `CongestionTimeFrame` parameter. Valid values range from `0x0` to `0x10000` (approximately 4000 requests).

Parameter Name	Default Value	Description
`CongestionTimeFrame`	`0x0`	A sliding window that defines the time period that is evaluated in seconds. If this parameter is not defined or is set to `0x0`, the path congestion feature is disabled because no time frame has been defined. Valid values range from `0x1` to `0x1C20` (approximately two hours).
`CongestionSamplingInterval`	`0x0`	The number of I/O requests that must be sent to a path before the *n*th request is used in the average response time calculation. For example, if this parameter is set to `100`, every 100th request sent to a path will be used in the average response time calculation. If this parameter is set to `0x0` or not defined, the path congestion feature is disabled for performance reasons—every I/O request would incur a calculation. Valid values range from `0x1` to `0xFFFFFFFF` (approximately 4 billion requests).
`CongestionMinPopulationSize`	`0x0`	The number of sampled I/O requests that must be collected before the average response time is calculated. Valid values range from `0x1` to `0xFFFFFFFF` (approximately 4 billion requests).
`CongestionTakeLastPathOffline`	`0x0`	A Boolean value that indicates whether the DSM driver will take the last path available to the storage array offline if the congestion thresholds have been exceeded. If this parameter is not defined or is set to `0x0`, the value is false. Valid values are `0x0` or `0x1`. **NOTE** Setting a path offline with the dsmUtil utility succeeds regardless of the setting of this value.

dsmUtil Utility

The dsmUtil utility is a command-line driven utility that works only with the Multipath I/O (MPIO) Device Specific Module (DSM) solution. The utility is used primarily as a way to instruct the DSM driver to perform various maintenance tasks, but the utility can also serve as a troubleshooting tool when necessary.

To use the dsmUtil utility, type this command, and press Enter:

```
dsmUtil [[-a [target_id]]
[-c array_name | missing]
[-d debug_level] [-e error_level] [-g virtual_target_id]
[-o [[feature_action_name[=value]] | [feature_variable_name=value]][,
SaveSettings]] [-M]
[-P [GetMpioParameters | MpioParameter=value | ...]] [-R]
[-s "failback" | "avt" | "busscan" | "forcerebalance"]
[-w target_wwn, controller_index]
```

NOTE The quotation marks must surround the parameters.

Typing `dsmUtil` without any parameters shows the usage information.

The following table shows the dsmUtil parameters.

Table 9 dsmUtil Parameters

Parameter	Description
`-a [target_id]`	Shows a summary of all storage arrays seen by the DSM. The summary shows the *`target_id`*, the storage array WWID, and the storage array name. If *`target_id`* is specified, DSM point-in-time state information appears for the storage array. On UNIX operating systems, the virtual HBA specifies unique target IDs for each storage array. The Windows MPIO virtual HBA driver does not use target IDs. The parameter for this option can be viewed as an offset into the DSM information structures, with each offset representing a different storage array. **NOTE** For use by Sun Customer Care Center representatives only.
`-c array_name | missing`	Clears the WWN file entries. This file is located in the `Program Files\DSMDrivers\mppdsm\WWN_FILES` directory with the extension `.wwn`. If the `array_name` keyword is specified, the WWN file for the specific storage array is deleted. If the `missing` keyword is used, all WWN files for previously attached storage arrays are deleted. If neither keyword is used, all of the WWN files, for both currently attached and previously attached storage arrays, are deleted.
`-d` *`debug_level`*	Sets the current debug reporting level. This option only works if the RDAC driver has been compiled with debugging enabled. Debug reporting is comprised of two segments. The first segment refers to a specific area of functionality, and the second segment refers to the level of reporting within that area. The *`debug_level`* is one of these hexadecimal numbers: ■ `0x20000000` – Shows messages from the RDAC driver's initialization routine. ■ `0x10000000` – Shows messages from the RDAC driver's device discovery routine. ■ `0x08000000` – Shows messages from the RDAC driver's ioctl() routine. ■ `0x04000000` – Shows messages from the RDAC driver's device open routine (Linux platforms only). ■ `0x02000000` – Shows messages from the RDAC driver's device read routine (Linux platforms only). ■ `0x01000000` – Shows messages related to HBA commands. ■ `0x00800000` – Shows messages related to aborted commands. ■ `x00400000` – Shows messages related to panic dumps. ■ `0x00200000` – Shows messages related to synchronous I/O activity. ■ `0x00100000` – Shows messages related to failover activity. ■ `0x00080000` – Shows messages related to failback activity. ■ `0x00040000` – Shows additional messages related to failback activity. ■ `0x00010000` – Shows messages related to device removals. ■ `0x00001000` – Shows messages related to SCSI reservation activity. ■ `0x00000400` – Shows messages related to path validation activity. ■ `0x00000001` – Debug level 1. ■ `0x00000002` – Debug level 2. ■ `0x00000004` – Debug level 3. ■ `0x00000008` – Debug level 4. You can combine these options with the logical or operator to provide multiple areas and levels of reporting as needed. **NOTE** For use by Sun Customer Care Center representatives only.

Parameter	Description
`-e` *`error_level`*	Sets the current error reporting level to *`error_level`*, which can have one of these values: ■ `0` – Show all errors. ■ `1` – Show path failover, controller failover, retryable, fatal, and recovered errors. ■ `2` – Show path failover, controller failover, retryable, and fatal errors. ■ `3` – Show path failover, controller failover, and fatal errors. This is the default setting. ■ `4` – Show controller failover and fatal errors. ■ `5` – Show fatal errors. For use by Sun Customer Care Center representatives only.
`-g` *`target_id`*	Displays detailed information about the state of each controller, path, and LUNs for the specified storage array. You can find the *`target_id`* by running the `dsmUtil -a` command.
`-M`	Shows the MPIO disk-to-drive mappings for the DSM. The output is similar to that found with the SMdevices utility. For use by Sun Customer Care Center representatives only.
`-o [[`*`feature_action_name`*`[=`*`value`*`]] | [`*`feature_variable_name=value`*`]][, SaveSettings]`	Troubleshoots a feature or changes a configuration setting. Without the `SaveSettings` keyword, the changes only affect the in-memory state of the variable. The `SaveSettings` keyword changes both the in-memory state and the persistent state. Some example commands are: ■ `dsmUtil -o` – Displays all the available feature action names. ■ `dsmUtil -o DisableLunRebalance=0x3` – Turns off the DSM-initiated storage array LUN rebalance (affects only the in-memory state).
`-P [GetMpioParameters | MpioParameter=`*`value`*` | ...]`	Displays and sets MPIO parameters. **NOTE** For use by Sun Customer Care Center representatives only.
`-R`	Remove the load-balancing policy settings for inactive devices.
`-s ["failback" | "avt" | "busscan" | "forcerebalance"]`	Manually initiates one of the DSM driver's scan tasks. A "failback" scan causes the DSM driver to reattempt communications with any failed controllers. An "avt" scan causes the DSM driver to check whether AVT has been enabled or disabled for an entire storage array. A "busscan" scan causes the DSM driver to go through its unconfigured devices list to see if any of them have become configured. A "forcerebalance" scan causes the DSM driver to move storage array volumes to their preferred controller and ignores the value of the `DisableLunRebalance` configuration parameter of the DSM driver.
`-w` *`target_wwn`*`, controller_index`	For use by Sun Customer Care Center representatives only.

Device Manager

Device Manager is part of the Windows operating system. Select Control Panel from the **Start** menu. Then select **Administrative Tools** >> **Computer Management** >> **Device Manager**.

The Device Manager tree for MPIO is similar to the one for RDAC. One difference is the names that are associated with the volumes. In RDAC, the volumes are identified with a label, such as the RDAC Volume. In MPIO, the volumes are named based on the vendor information and product ID information of the underlying physical device, along with the text Multi-Path Disk Device.

Scroll down to System Devices to view information about the DSM driver itself. This name might be different based on your network configuration.

The Drives section shows both the drives identified with the HBA drivers and the volumes created by MPIO. Select one of the MPIO volumes, and right-click it. Select **Properties** to open the Multi-Path Disk Device Properties window.

This properties window shows if the device is working correctly. Select the Driver tab to view the driver information.

Determining if a Path Has Failed

If a single path to a controller that has multiple paths fails, the failover driver makes an entry in the OS system log that indicates a path failure. In the storage management software, the storage array shows a Degraded status.

If all of the paths to a controller fail, the failover driver makes entries in the OS system log that indicate a path failure and failover. In the storage management software, the storage array shows a Needs Attention condition of Volume not on preferred path. The failover event is also written to the Major Event Log (MEL). In addition, if the administrator has configured alert notifications, email messages, or SNMP traps, messages are posted for this condition. The Recovery Guru in the storage management software provides more information about the path failure, along with instructions about how to correct the problem.

NOTE Alert reporting for the Volume not on preferred path condition can be delayed if you set the failover alert delay parameter through the storage management software. When you set this parameter, it imposes a delay on the setting of the Needs Attention condition in the storage management software.

Frequently Asked Questions About Windows Failover Drivers

Table 10 Frequently Asked Questions about Windows Failover Drivers

Question	Answer
My disk devices or host bus adapters (HBAs) show a yellow exclamation point. What does this mean?	When you use Device Manager, you might observe that a disk device icon or an HBA icon has a yellow exclamation point on it. If new volumes have been mapped to the host, the exclamation point might appear on the icon for a few seconds. This action occurs because the PnP Manager is reconfiguring the device, and, during this time, the device or the HBA might not be used. If the exclamation point stays for more than one minute, a configuration error has occurred.
My disk devices or HBAs show a red X. What does this mean?	When you use Device Manager, you might notice that a disk device icon or an HBA icon has a red X on it. This X indicates that the device has been disabled. A disabled device cannot be used or communicated with until it is re-enabled. If the disabled device is an adapter, any disk devices that were connected to that adapter are removed from Device Manager.
Why does the SMdevices utility not show any volumes?	If the SMdevices utility does not show any volumes, perform these steps: 1. Make sure that all cables are seated correctly. Make sure that all gigabit interface converters (GBICs) are seated correctly. 2. Determine the HBA BIOS and driver versions that the system uses, and make sure that the HBA BIOS and driver versions are correct. 3. Make sure that your mappings are correct. Do not use any HBA mapping tools. 4. Use WinObj to determine if the host has detected the volumes. If the host has not detected the volumes, an HBA problem or a controller problem has occurred. Make sure that the HBAs are logging into the switch or the controller. If they are not logging in, the problem is probably HBA related. If the HBAs have logged into the controller, a controller issue might be the problem.
The SMdevices utility shows duplicate entries for some or all of my disks.	You see that some of your disks show up twice when you run the SMdevices utility. For the Windows Server OS, something went wrong with the device-claiming process.
I run the hot_add utility, but my disks do not appear.	See “Why does the SMdevices utility not show any volumes?
I have mapped new volumes to my host, but I cannot see them.	Run the hot_add utility. See “Why does the SMdevices utility not show any volumes?”

Question	Answer
How do I know if a host has detected my volumes?	Use WinObj to determine if the host can see the volumes. ■ If the host cannot see the volumes, an HBA problem or a controller problem has occurred. ■ Make sure that the HBAs log into the switch or the controller. If they are not logging in correctly, the problem is probably HBA related. ■ If the HBAs have logged into the controller, the problem might be a controller issue.
When I boot my system, I get a “Registry Corrupted” message.	Refer to the Microsoft Knowledge Base article 277222 at http://support.microsoft.com/kb/277222/en-us. Registry limitations can result in devices and paths that are not recognizable by the host OS and the failover driver.
My controller failover test does not fail over.	Make sure that you have looked through the rest of this document for the problem. If you think that the problem is still RDAC-related or DSM-related, contact a Sun Customer Care Center representative.
After I install the DSM driver, my system takes a long time to start. Why?	You might still experience long start times after you install the DSM driver because the Windows OS is completing its configuration for each device. For example, you install the DSM driver on a host with no storage array attached, and you restart the host. Before the Windows OS actually starts, you plug in a cable to a storage array with 32 volumes. In the start-up process, PnP detects the configuration change and starts to process it. After the configuration change has completed, subsequent restarts do not experience any delays unless additional configuration changes are detected. The same process can occur even if the host has already started.
What host type must I use for the MPIO solution?	If you use Microsoft Cluster Server, select a host type of the Windows 2003/2008 Clustered OS. If you do not use Microsoft Cluster Server, select a host type of the Windows 2003/2008 Non-Clustered OS.
How can I tell if MPIO is installed?	Perform these steps: 1. Go to the **Control Panel** on the **Start** menu, and double-click **Administrative Tools**. 2. Select **Computer Management >> Device Manager >> SCSI and RAID controllers**. 3. On Windows Server 2003, look for Multi-Path Support. On Windows Server 2008, look for Microsoft Multi-Path Bus Driver. If one of these items is present, MPIO is installed.

Question	Answer
How can I tell if the DSM driver is installed?	Perform these steps: 1. Go to the **Control Panel** on the **Start** menu, and double-click **Administrative Tools**. 2. Select **Computer Management** >> **Device Manager** >> **System Devices**. 3. Look for the LSI-supported DSM. The name ends with the text Device-Specific Module for Multi-Path. If it is present, DSM is installed.
What is the default vendor ID string and the product ID string?	By default, the vendor ID string and the product ID string configured for LSI storage arrays are named LSI/INF-01-00. If not, the PnP manager cannot choose the failover driver to manage the volume. The driver takes over, which causes delays. If you suspect that this event has occurred, check the non-user configuration region of the controller firmware

What should I do if I receive this message?
`Warning: Changing the storage array name can cause host applications to lose access to the storage array if the host is running certain path failover drivers.`
`If any of your hosts are running path failover drivers, please update the storage array name in your path failover driver’s configuration file before rebooting the host machine to insure uninterrupted access to the storage array. Refer to your path failover driver documentation for more details.`
	You do not need to update files. The information is dynamically created only when the storage array is found initially. Use one of these two options to correct this behavior:
	■ Restart the host server.
	■ Unplug the storage array from the host server, and perform a rescan of all of the devices. After the devices have been removed from the storage array, you can re-attach them. Another rescan takes place, which rebuilds the information with the updated names.

Installing or Upgrading SANtricity ES and DSM on the Windows OS

NOTE For SANtricity ES Storage Manager 10.75 on the Windows OS, only the Microsoft Multipath I/O (MPIO) Device Specific Module (DSM) failover driver is supported. You cannot install the DSM driver and the RDAC driver on the same system at the same time.

Perform the steps in this task to install the SANtricity ES Storage Manager and DSM or to upgrade from an earlier release of the SANtricity ES Storage Manager and DSM on a system with a Windows operating system. For a clustered system, perform these steps on each node of the system, one node at a time.

1. Open the installation program on the SANtricity ES Storage Manager Installation DVD.

   The SANtricity ES Storage Manager installation window appears.

2. Click **Next**.
3. Accept the terms of the license agreement, and click **Next**.
4. Select **Custom**, and click **Next**.
5. Select the applications that you want to install.
   a. Click the name of an application to see its description.
   b. Select the check box next to an application to install it.

6. Click **Next**.

   If you have a previous version of the software installed, you will receive a warning message:

   ```
   Existing versions of the following software already reside on
   this computer ... If you choose to continue, the existing
   versions will be overwritten with new versions ....
   ```
7. If you receive this warning and want to update to SANtricity ES Storage Manager Version 10.75, click **OK**.
8. Select whether to automatically start the Event Monitor. Click **Next**.
   - Start the Event Monitor for the one I/O host on which you want to receive alert notifications.
   - Do not start the Event Monitor for all other I/O hosts attached to the storage array or for computers that you use to manage the storage array.
9. Click **Next**.
10. If you receive a warning about antivirus or backup software that is installed, click **Continue**.
11. Read the pre-installation summary, and click **Install**.
12. Wait for installation to complete, and click **Done**.

Removing SANtricity ES and DSM from the Windows OS

NOTE To prevent loss of data, the host from which you are removing SANtricity ES Storage Manager and the DSM must have only one path to the storage array. Reconfigure the connections between the host and the storage array to remove any redundant connections before you uninstall SANtricity ES Storage Manager and the DSM.

1. From the Windows **Start** menu, select **Control Panel**.

   The Control Panel window appears.
2. In the Control Panel window, double-click **Add or Remove Programs**.

   The Add or Remove Programs window appears.
3. Select **SANtricity ES Storage Manager**.
4. Click the **Remove** button to the right of the **SANtricity ES Storage Manager** entry.

WinObj

You can use WinObj to view the Object Manager namespace that is maintained by the operating system. Every Windows OS driver that creates objects in the system can associate a name with the object that can be viewed from WinObj. With WinObj, you can view the volumes and paths that the host bus adapters (HBAs) have identified. You can also view what the failover driver identifies from a storage array.

Chapter 8: Failover Drivers for the Linux Operating System

Redundant Dual Active Controller (RDAC) is the supported failover driver for SANtricity ES Storage Manager with Linux operating systems.

Linux OS Restrictions

This version of the Linux OS RDAC does not support any Linux OS 2.4 kernels, such as the following:

- SUSE SLES 8 OS
- Red Hat 3 Linux OS
- SLES 8 and Red Hat 3 Linux OSs on POWER (LoP) servers

The Linux OS RDAC driver cannot coexist with a Fibre Channel host bus adapter (HBA)-level multi-path failover or failback driver, such as these:

- The 8.00.00-fo or 8.00.02-fo Linux OS device driver for the IBM DS4000 fc2-133 host bus adapters driver on servers with Intel architecture processors or AMD architecture processors
- The QLA driver for the Fibre Channel expansion adapters on the IBM LoP blade servers

You might have to modify the makefile of the HBA driver for it to be compiled in the non-failover mode.

Auto-Volume Transfer (AVT) mode is automatically enabled in the Storage Domains host type in the Linux OS. This mode causes contention when RDAC is installed, so you must disable it before you install RDAC. Also, there is a separate Linux OS host type in which AVT is already disabled.

When the RDAC driver detects that all paths to a storage array have failed, it reports I/O failure immediately. This behavior is different from the behavior of the failover device driver from IBM for the Fibre Channel HBA. The Fibre Channel HBA failover device driver waits for a certain timeout or retry period before reporting an I/O failure to the host application.

The cluster support feature in the Linux OS RDAC driver is only available for controller firmware version 5.4*x.xx.xx* or later. If a SCSI-2 reserve command or a SCSI-2 release command is addressed to a volume on a storage array that runs a firmware version earlier than 5.40, a check condition is returned.

LoP servers do not support clustering with this Linux OS RDAC.

For LoP servers, the `modprobe` command is not supported with this Linux OS RDAC release on SUSE SLES 9 when the Linux OS RDAC driver is installed. Use the `insmod` command to remove and recover the device driver stack. On LoP pSeries servers with more than three processors, if you use the `modprobe` command, it might result in a hung server or panics.

Do not assign a universal access volume to LUN ID 0, especially with RDAC installed. If you place a universal access volume at LUN ID 0, it might lead to loss of volume recognition of next sequence LUN IDs or a partial list of virtual volumes reported with the RDAC driver. Assign these volumes to LUN ID 31.

For LoP servers, the HBA hot-swap procedure to remove a live, fully functioning Fibre Channel HBA causes a system panic if the I/O is not diverted from that path first. This functionality is only supported on current 2.6 kernel versions. The work-around is to pull the Fibre Channel cable first, wait two minutes to five minutes for failover to complete, and then run the `drslot_chrp_pci -r -s slot-number` command.

The Linux operating system includes a new method of kernel dump (kdump/kexec) to replace the previous LKCD (Linux Kernel Crash Dump) method for SUSE and diskdump method for RHEL. If kdump is configured, kdump and kexec work together to capture kernel exceptions, save the kernel state/memory and start a second kernel image for debugging the troubled kernel. The second kernel is called kdump kernel. Kexec and kdump are useful for troubleshooting panic conditions. With the current installation and driver build method, the RDAC driver modules are not included in the kdump kernel initrd image. An initrd image contains all necessary device driver modules that are used after the kernel image is loaded and before user space initialization is started. Because the RDAC driver modules are not included in the kdump initrd image, a user experiences these problems when kdump is configured: long boot times and the inability to save vmcore file in a SAN boot configuration. The RDAC driver installation will build a kdump kernel initrd image with the RDAC driver modules inside. Also, when the RDAC driver is installed, it will detect that a kdump kernel is configured, and update the initrd image with the RDAC drivers.

Unique Features of RDAC from LSI

Redundant Dual Active Controller is the failover driver for the Linux OS that is included in SANtricity ES Storage Manager. The RDAC failover driver includes these unique features:

- On-the-fly path validation.
- Cluster support.
- Automatic detection of path failure. The RDAC failover driver automatically routes I/O to another path in the same controller or to an alternate controller, in case all paths to a particular controller fail.
- Retry handling is improved, because the RDAC driver can better understand the controller-returned sense key/ASC/ASCQ of vendor-specific statuses of LSI.
- Automatic rebalance is handled. When the failed controller obtains Optimal status, storage array rebalance is performed automatically without user intervention.
- Three load-balancing policies are supported: round robin subset, least queue depth, and path weight.

Configuration Settings for Windows DSM and Linux RDAC

This topic applies to both the Windows OS and the Linux OS. The failover driver that is provided with the storage management software contains configuration settings that can modify the behavior of the driver.

- For the Linux OS, the configuration settings are in the `/etc/mpp.conf` file.
- For the Windows OS, the configuration settings are in the `HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\<DSM_Driver>\Parameters` registry key, where `<DSM_Driver>` is the name of the OEM-specific driver

The default driver is `mppdsm.sys`. Any changes to the settings take effect on the next reboot of the host.

The default values listed in the following table apply to both Windows and Linux unless the OS is specified in parentheses. Many of these values are overridden by the failover installer for the Linux OS or the Windows OS.

ATTENTION Possible loss of data access – If you change these settings from their configured values, you might lose access to the storage array.

Table 11 Configuration Settings for Windows DSM and Linux RDAC

Parameter Name	Default Value (Operating System)	Description
`MaxPathsPerController`	`4`	The maximum number of paths (logical endpoints) that are supported per controller. The total number of paths to the storage array is the `MaxPathsPerController` value multiplied by the number of controllers. The allowed values range from `0x1` (1) to `0x20` (32) for Windows, and from `0x1` (1) to `0xFF` (255) for Linux RDAC. For use by Sun Customer Care Center representatives only.
`ScanInterval`	`1` (Windows) `60` (Linux)	The interval time, in seconds, that the failover driver will check for these conditions: ■ A change in preferred ownership for a LUN ■ An attempt to rebalance LUNs to their preferred paths ■ A change in AVT enabled status or disabled status For the Windows OSs, the allowed values range from `0x1` to `0xFFFFFFFF` and must be specified in minutes. For the Linux OSs, the allowed values range from `0x1` to `0xFFFFFFFF` and must be specified in seconds. For use by Sun Customer Care Center representatives only.
`ErrorLevel`	`3`	This setting determines which errors to log. These values are valid: ■ `0` – Display all errors ■ `1` – Display path failover errors, controller failover errors, retryable errors, fatal errors, and recovered errors ■ `2` – Display path failover errors, controller failover errors, retryable errors, and fatal errors ■ `3` – Display path failover errors, controller failover errors, and fatal errors ■ `4` – Display controller failover errors, and fatal errors For use by Sun Customer Care Center representatives only.
`SelectionTimeoutRetryCount`	`0`	The number of times a selection timeout is retried for an I/O request before the path fails. If another path to the same controller exists, the I/O is retried. If no other path to the same controller exists, a failover takes place. If no valid paths exist to the alternate controller, the I/O is failed. The allowed values range from `0x0` to `0xFFFFFFFF`. For use by Sun Customer Care Center representatives only.

Parameter Name	Default Value (Operating System)	Description
`CommandTimeoutRetryCount`	`1`	The number of times a command timeout is retried for an I/O request before the path fails. If another path to the same controller exists, the I/O is retried. If another path to the same controller does not exist, a failover takes place. If no valid paths exist to the alternate controller, the I/O is failed. The allowed values range from `0x0` to `0xa` (10) for Windows, and from `0x0` to `0xFFFFFFFF` for Linux RDAC. For use by Sun Customer Care Center representatives only.
`UaRetryCount`	`10`	The number of times a Unit Attention (UA) status from a LUN is retried. This parameter does not apply to UA conditions due to Quiescence in Progress. The allowed values range from `0x0` to `0x64` (100) for Windows, and from `0x0` to `0xFFFFFFFF` for Linux RDAC. For use by Sun Customer Care Center representatives only.
`SynchTimeout`	`120`	The timeout, in seconds, for synchronous I/O requests that are generated internally by the failover driver. Examples of internal requests include those related to rebalancing, path validation, and issuing of failover commands. The allowed values range from `0x1` to `0xFFFFFFFF`. For use by Sun Customer Care Center representatives only.
`DisableLunRebalance`	`0`	This parameter provides control over the LUN failback behavior of rebalancing LUNs to their preferred paths. These values are possible: ■ `0` – LUN rebalance is enabled for both AVT and non-AVT modes. ■ `1` – LUN rebalance is disabled for AVT mode and enabled for non-AVT mode. ■ `2` – LUN rebalance is enabled for AVT mode and disabled for non-AVT mode. ■ `3` – LUN rebalance is disabled for both AVT mode and non-AVT mode. ■ `4` – The selective LUN Transfer feature is enabled if AVT mode is off and `ClassicModeFailover` is set to LUN level 1.
`S2ToS3Key`	Unique key	This value is the SCSI-3 reservation key generated during failover driver installation. For use by Sun Customer Care Center representatives only.
`LoadBalancePolicy`	`1`	This parameter determines the load-balancing policy used by all volumes managed by the Windows DSM and Linux RDAC failover drivers. These values are valid: ■ `0` – Round robin with subset. ■ `1` – Least queue depth with subset. ■ `2` – Least path weight with subset (Windows OS only).

Parameter Name	Default Value (Operating System)	Description
`ClassicModeFailover`	`0`	This parameter provides control over how the DSM handles failover situations. These values are valid: ■ `0` – Perform controller-level failover (all LUNs are moved to the alternate controller). ■ `1` – Perform LUN-level failover (only the LUNs indicating errors are transferred to the alternate controller).
`SelectiveTransferMaxTransferAttempts`	`3`	This parameter sets the maximum number of times that a host will transfer the ownership of a LUN to the alternate controller when the Selective LUN Transfer mode is enabled. This setting prevents multiple hosts from continually transferring LUNs between controllers.
`SelectiveTransferMinIOWaitTime`	`5`	This parameter sets the minimum wait time (in seconds) that the DSM will wait before transferring a LUN to the alternate controller when the Selective LUN Transfer mode is enabled. This parameter tries to stop excessive LUN transfers due to intermittent link errors.

Prerequisites for Installing RDAC on the Linux OS

Before installing RDAC on the Linux operating system, make sure that your storage array meets these conditions:

- Make sure that the host system on which you want to install the RDAC driver has supported HBAs.
- Refer to the installation electronic document topics for your controller module or array module for any configuration settings that you need to make.
- Although the system can have Fibre Channel HBAs from multiple vendors or multiple models of Fibre Channel HBAs from the same vendor, you can connect only the same model of Fibre Channel HBAs to each storage array.
- Make sure that the low-level HBA driver has been correctly built and installed before RDAC driver installation.
- The standard HBA driver must be loaded before you install the RDAC driver. The HBA driver has to be a non-failover driver.
- For LSI HBAs, the port driver is named `mptbase`, and the host driver is named `mptscsi` or `mptscsih`, although the name depends on the driver version. The Fibre Channel driver is named `mptfc`, the SAS driver is named `mptsas`, and the SAS2 driver is named `mpt2sas`.
- For QLogic HBAs, the base driver is named `qla2xxx`, and host driver is named `qla2300`. The 4-GB HBA driver is named `qla2400`.
- For IBM Emulex HBAs, the base driver is named `lpfcdd` or `lpfc`, although the name depends on the driver version.
- For Emulex HBAs, the base driver is named `lpfcdd` or `lpfc`, although the name depends on the driver version.
- Make sure that the kernel source tree for the kernel version to be built against is already installed. You must install the kernel source rpm on the target system for the SUSE SLES operating system. You are not required to install the kernel source for the Red Hat operating system.
- Make sure that the necessary kernel packages are installed: `source rpm` for the SUSE Linux Enterprise Server operating system and `kernel headers/kernel devel` for the Red Hat Enterprise Linux operating system.

In SUSE operating systems, you must include these items for the HBAs mentioned as follows:

- For LSI HBAs, INITRD_MODULES includes `mptbase` and `mptscsi` (or `mptscsih`) in the `/etc/sysconfig/kernel` file. The Fibre Channel driver is named `mptfc`, the SAS driver is named `mptsas`, and the SAS2 driver is named `mpt2sas`.
- For QLogic HBAs, INITRD_MODULES includes a `qla2xxx` driver and a `qla2300` driver in the `/etc/sysconfig/kernel` file.
- For IBM Emulex HBAs, INITRD_MODULES includes an `lpfcdd` driver or an `lpfc` driver in the `/etc/sysconfig/kernel` file.
- For Emulex HBAs, INITRD_MODULES includes an `lpfcdd` driver or an `lpfc` driver in the `/etc/sysconfig/kernel` file.

Installing SANtricity ES Storage Manager and RDAC on the Linux OS

NOTE SANtricity ES Storage Manager requires that the different Linux OS kernels have separate installation packages. Make sure that you are using the correct installation package for your particular Linux OS kernel.

1. Open the installation program on the SANtricity ES Storage Manager Installation DVD.

   The SANtricity ES Storage Manager installation window appears.
2. Click **Next**.
3. Accept the terms of the license agreement, and click **Next**.
4. Select one of the installation packages:
   - **Typical** – Select this option to install all of the available host software.
   - **Management Station** – Select this option to install software to configure, manage, and monitor a storage array. This option does not include RDAC. This option only installs the client software.
   - **Host** – Select this option to install the storage array server software.
   - **Custom** – Select this option to customize the features to be installed.

NOTE For this procedure, **Typical** is selected. If the **Host** installation option is selected, the Agent, the Utilities, and the RDAC driver will be installed.

   You might receive a warning after you click **Next**. The warning states:

```
Existing versions of the following software already reside on
this computer ... If you choose to continue, the existing
versions will be overwritten with new versions ....
```

   If you receive this warning and want to update to SANtricity ES Storage Manager Version 10.75, click **OK**.

5. Click **Install**.

   You will receive a warning after you click **Install**. The warning tells you that the RDAC driver is not automatically installed. You must manually install the RDAC driver.

   The RDAC source code is copied to the specified directory in the warning message. Go to that directory, and perform the steps in Installing RDAC Manually on the Linux OS.
6. Click **Done**.

Installing RDAC Manually on the Linux OS

1. To unzip the RDAC `tar.gz` file and untar the RDAC tar file, type this command, and press Enter:

```
tar -zxvf <filename>
```

2. Go to the Linux RDAC directory.

3. Type this command, and press Enter.

   ```
   make uninstall
   ```

4. To remove the old driver modules in that directory, type this command, and press Enter:

   ```
   make clean
   ```

5. To compile all driver modules and utilities in a multiple CPU server (SMP kernel), type this command, and press Enter:

   ```
   make
   ```

6. Type this command, and press Enter:

   ```
   make install
   ```

   These actions result from running this command:

   — The driver modules are copied to the kernel module tree.

   — The new RAMdisk image (`mpp-`uname -r`.img`) is built, which includes the RDAC driver modules and all driver modules that are needed at boot.

7. Follow the instructions shown at the end of the build process to add a new boot menu option that uses `/boot/mpp-`uname -r`.img` as the initial RAMdisk image.

Making Sure that RDAC Is Installed Correctly on the Linux OS

1. Restart the system by using the new boot menu option.
2. Make sure that these driver stacks were loaded after restart:

   — `scsi_mod`

   — `sd_mod`

   — `sg`

   — `mppUpper`

   — The physical HBA driver module

   — `mppVhba`

3. Type this command, and press Enter:

   ```
   /sbin/lsmod
   ```

4. To make sure that the RDAC driver discovered the available physical volumes and created virtual volumes for them, type this command, and press Enter:

   ```
   /opt/mpp/lsvdev
   ```

   You can now send I/O to the volumes.

5. If you make any changes to the RDAC configuration file (`/etc/mpp.conf`) or the persistent binding file (`/var/mpp/devicemapping`), run the `mppUpdate` command to rebuild the RAMdisk image to include the new file. In this way, the new configuration file (or persistent binding file) can be used on the next system restart.
6. To dynamically reload the driver stack (`mppUpper`, physical HBA driver modules, `mppVhba`) without restarting the system, perform these steps:

   a. To unload the `mppVhba` driver, type this command, and press Enter:

      ```
      rmmod mppVhba
      ```

   b. To unload the physical HBA driver, type this command, and press Enter:

      ```
      modprobe -r "physical hba driver modules"
      ```

   c. To unload the `mppUpper` driver, type this command, and press Enter:

      ```
      rmmod mppUpper
      ```

   d. To reload the `mppUpper` driver, type this command, and press Enter:

      ```
      modprobe mppUpper
      ```

e. To reload the physical HBA driver, type this command, and press Enter:

```
modprobe "physical hba driver modules"
```

f. To reload the mppVhba driver, type this command, and press Enter:

```
modprobe mppVhba
```

7. Restart the system whenever there is an occasion to unload the driver stack.

NOTE Using the modprobe command with the RDAC driver stack or using the rmmod command to remove all the drivers in the RDAC driver stack, in order, is not recommended nor supported.

8. Disable Auto-Volume Transfer (AVT) by issuing a set controller command line interface (CLI) command. For example, to disable AVT for host region 6 and controller A, use this command:

```
set controller[a] hostNVSRAMByte[0x6, 0x24]=0x00;
```

9. Use a utility, such as devlabel, to create user-defined device names that can map devices based on a unique identifier, called a UUID.
10. Use the udev command for persistent device names. The udev command dynamically generates device name links in the /dev/disk directory based on path, ID or UUID.

```
linux-kbx5:/dev/disk # ls /dev/disk by-id  by-path  by-uuid
```

For example, the /dev/disk/by-id directory links volumes that are identified by WWIDs of the volumes to actual disk device nodes.

```
lrwxrwxrwx 1 root root 10 Feb 23 12:15
scsi-3600a0b80000c2df9000003b141417799 -> ../../sdda
lrwxrwxrwx 1 root root  9 Feb 23 12:15
scsi-3600a0b80000f27030000000d416b94fd -> ../../sdc
lrwxrwxrwx 1 root root  9 Feb 23 12:15
scsi-3600a0b80000f270300000015416b958f -> ../../sdg
```

Configuring Failover Drivers for the Linux OS

The Windows OS and the Linux OS share the same set of tunable parameters to enforce the same I/O behaviors. For a description of these parameters, go to Configuration Settings for Windows DSM and Linux RDAC on page 33.

Parameter Name	Default Value	Description
`ImmediateVirtLunCreate`	`0`	This parameter determines whether to create the virtual LUN immediately if the owning physical path is not yet discovered. This parameter can take the following values: ■ `0` – Do not create the virtual LUN immediately if the owning physical path is not yet discovered. ■ `1` – Create the virtual LUN immediately if the owning physical path is not yet discovered.
`BusResetTimeout`		The time, in seconds, for the RDAC driver to delay before retrying an I/O operation if the DID_RESET status is received from the physical HBA. A typical setting is `150`.
`AllowHBAsgDevs`	`0`	This parameter determines whether to create individual SCSI generic (SG) devices for each I:T:L for the end LUN through the physical HBA. This parameter can take the following values: ■ `0` – Do not allow creation of SG devices for each I:T:L through the physical HBA. ■ `1` – Allow creation of SG devices for each I:T:L through the physical HBA.

Compatibility and Migration

Controller firmware – The Linux OS RDAC driver is compatible with the controller firmware. However, the Linux OS RDAC driver does not support SCSI-2 to SCSI-3 reservation translation unless the release is version 8.40.*xx* or later.

Linux OS distributions – The Linux OS RDAC driver is intended to work on any Linux OS distribution that has the standard SCSI I/O storage array (SCSI middle-level and low-level interfaces). This release is targeted specifically at SUSE Linux OS Enterprise Server and Red Hat Advanced Server.

mppUtil Utility

The mppUtil utility is a general-purpose command-line driven utility that works only with MPP-based RDAC solutions. The utility instructs RDAC to perform various maintenance tasks but also serves as a troubleshooting tool when necessary.

To use the mppUtil utility, type this command, and press Enter:

```
mppUtil [-a target_name] [-c wwn_file_name] [-d debug_level]
[-e error_level] [-g virtual_target_id] [-I host_num]
[-o feature_action_name[=value][, SaveSettings]]
[-s "failback" | "avt" | "busscan" | "forcerebalance"] [-S] [-U]
[-V] [-w target_wwn,controller_index]
```

NOTE The quotation marks must surround the parameters.

The mppUtil utility is a cross-platform tool. Some parameters might not have a meaning in a particular operating system environment. A description of each parameter follows.

Table 12 mppUtil Parameters

Parameter	Description
`-a` *`target_name`*	Shows the RDAC driver's internal information for the specified virtual *`target_name`* (storage array name). If a *`target_name`* value is not included, the `-a` parameter shows information about all of the storage arrays that are currently detected by this host.
`-c` *`wwn_file_name`*	Clears the WWN file entries. This file is located at `/var/mpp` with the extension `.wwn`.
`-d` *`debug_level`*	Sets the current debug reporting level. This option works only if the RDAC driver has been compiled with debugging enabled. Debug reporting is comprised of two segments. The first segment refers to a specific area of functionality, and the second segment refers to the level of reporting within that area. The *`debug_ level`* is one of these hexadecimal numbers: ■ `0x20000000` – Shows messages from the RDAC driver's init() routine. ■ `0x10000000` – Shows messages from the RDAC driver's attach() routine. ■ `0x08000000` – Shows messages from the RDAC driver's ioctl() routine. ■ `0x04000000` – Shows messages from the RDAC driver's open() routine. ■ `0x02000000` – Shows messages from the RDAC driver's read() routine. ■ `0x01000000` – Shows messages related to HBA commands. ■ `0x00800000` – Shows messages related to aborted commands. ■ `0x00400000` – Shows messages related to panic dumps. ■ `0x00200000` – Shows messages related to synchronous I/O activity. ■ `0x00000001` – Debug level 1. ■ `0x00000002` – Debug level 2. ■ `0x00000004` – Debug level 3. ■ `0x00000008` – Debug level 4. These options can be combined with the logical and operator to provide multiple areas and levels of reporting as needed. For use by Sun Customer Care Center representatives only.
`-e` *`error_level`*	Sets the current error reporting level to *`error_ level`*, which can have one of these values: ■ `0` – Show all errors. ■ `1` – Show path failover, controller failover, retryable, fatal, and recovered errors. ■ `2` – Show path failover, controller failover, retryable, and fatal errors. ■ `3` – Show path failover, controller failover, and fatal errors. This is the default setting. ■ `4` – Show controller failover and fatal errors. ■ `5` – Show fatal errors. For use by Sun Customer Care Center representatives only.
`-g` *`virtual_target_id`*	Display the RDAC driver's internal information for the specified *`virtual_target_id`*.
`-I` *`host_num`*	Prints the maximum number of targets that can be handled by that host. Here, host refers to the HBA drivers on the system and includes the RDAC driver. The host number of the HBA driver is given as an argument. The host numbers assigned by the Linux middle layer start from 0. If two ports are on the HBA card, host numbers 0 and 1 would be taken up by the low-level HBA driver, and the RDAC driver would be at host number 2. Use `/proc/scsi` to determine the host number.

Parameter	Description
`-o` *`feature_action_name`*`[=`*`value`*`][, SaveSettings]`	Troubleshoots a feature or changes a configuration setting. Without the `SaveSettings` keyword, the changes affect only the in-memory state of the variable. The `SaveSettings` keyword changes both the in-memory state and the persistent state. You must run `mppUpdate` to reflect these changes in the inird image before rebooting the server. Some example commands are: ■ `mppUtil -o` – Displays all the available feature action names. ■ `mppUtil -o ErrorLevel=0x2` – Sets the `ErrorLevel` parameter to `0x2` (affects only the in-memory state).
`-s ["failback" | "avt" | "busscan" | "forcerebalance"]`	Manually initiates one of the RDAC driver's scan tasks. ■ A "failback" scan causes the RDAC driver to reattempt communications with any failed controllers. ■ An "avt" scan causes the RDAC driver to check whether AVT has been enabled or disabled for an entire storage array. ■ A "busscan" scan causes the RDAC driver to go through its unconfigured devices list to see if any of them have become configured. ■ A "forcerebalance" scan causes the RDAC driver to move storage array volumes to their preferred controller and ignore the value of the `DisableLunRebalance` configuration parameter of the RDAC driver.
`-S`	Reports the Up state or the Down state of the controllers and paths for each LUN in real time.
`-U`	Refreshes the Universal Transport Mechanism (UTM) LUN information in MPP driver internal data structure for all the storage arrays that have already been discovered.
`-V`	Prints the version of the RDAC driver currently running on the system.
`-w` *`target_wwn`*`,`*`controller_index`*	For use by Sun Customer Care Center representatives only.

Frequently Asked Questions about Linux Failover Drivers

Table 13 Frequently Asked Questions about Linux Failover Drivers

Question	Answer
How do I get logs from RDAC in the Linux OS?	Use the `mppSupport` command to obtain several logs related to RDAC. The `mppSupport` command is found in the `/opt/mpp/mppSupport` directory. The command creates a file named `mppSupportdata_`*`hostname_RDAC version_datetime`*`.tar.gz` in the `/tmp` directory.
How does persistent naming work?	The Linux OS SCSI device names can change when the host system restarts. Use a utility, such as devlabel, to create user-defined device names that will map devices based on a unique identifier. The udev method is the preferred method for SLES 10 and RHEL.
What must I do after applying a kernel update?	After you apply the kernel update and start the new kernel, perform these steps to build the RDAC Initial Ram Disk image (initrd image) for the new kernel: 1. Change the directory to the Linux RDAC source code directory. 2. Type `make uninstall`, and press Enter. 3. Reinstall RDAC.

Question	Answer
What is the Initial Ram Disk Image (initrd image), and how do I create a new initrd image?	The initrd image is automatically created when the driver is installed by using the `make install` command. The boot loader configuration file must have an entry for this newly created image. The initrd image is located in the boot partition. The file is named `mpp'-uname -r'.img`. For a driver update, if the system already has a previous entry for RDAC, the system administrator must modify the existing RDAC entry in the boot loader configuration file. In most of the cases, no change is required if the kernel version is the same. To create a new initrd image, type `mppUpdate`, and press Enter. The old image file is overwritten with the new image file. If third-party drivers are needed to be added to the initrd image, change the `/etc/sysconfig/ kernel` file (SUSE) with the third-party driver entries. Run the `mppUpdate` command again to create a new initrd image.
How do I remove unmapped or disconnected devices from the existing host?	Run `hot_add -d` to remove all unmapped or disconnected devices.
What if I remap a LUN from the storage array?	Run `hot_add -u` to update the host with the changed LUN mapping.
What if I change the size of the LUN on the storage array?	Run `hot_add -c` to change the size of the LUN on the host.

How do I make sure that RDAC finds the available storage arrays?	To make sure that the RDAC driver has found the available storage arrays and created virtual storage arrays for them, type these commands, and press Enter after each command.
	`ls -lR /proc/mpp`
	`mppUtil -a`
	`/opt/mpp/lsvdev`
	To show all attached and discovered volumes, type`cat /proc/scsi/scsi`, and press Enter.

What should I do if I receive this message?
`Warning: Changing the storage array name can cause host applications to lose access to the storage array if the host is running certain path failover drivers.`
`If any of your hosts are running path failover drivers, please update the storage array name in your path failover driver's configuration file before rebooting the host machine to insure uninterrupted access to the storage array. Refer to your path failover driver documentation for more details.`
	The path failover drivers that cause this warning are the RDAC drivers on both the Linux OS and the Windows OS.
	The storage array user label is used for storage array-to-virtual target ID binding in the RDAC driver. For the Linux OS, change this file to add the storage array user label and its virtual target ID.
	`.~ # more /var/mpp/devicemapping`

Chapter 9: Device Mapper Multipath for the Linux Operating System

Device Mapper (DM) is a generic framework for block devices provided by the Linux operating system. It supports concatenation, striping, snapshots, mirroring, and multipathing. The multipath function is provided by the combination of the kernel modules and user space tools.

The DMMP is supported on SUSE Linux Enterprise Server (SLES) Version 11. The SLES installation must have components at or above the version levels shown in the following table before you install the DMMP.

Table 14 Minimum Supported Configurations for the SLES 11 Operating System

Version	Component
Kernel version	`kernel-default-2.6.27.29-0.1.1`
Scsi_dh_rdac kmp	`lsi-scsi_dh_rdac-kmp-default-0.0_2.6.27.19_5-1`
Device Mapper library	`device-mapper-1.02.27-8.6`
Multipath-tools	`multipath-tools-0.4.8-40.6.1`

To update a component, download the appropriate package from the Novell website at http://download.novell.com/patch/finder. The Novell publication, *SUSE Linux Enterprise Server 11 Installation and Administration Guide*, describes how to install and upgrade the operating system.

Device Mapper Features

- Provides a single block device node for a multipathed logical unit
- Ensures that I/O is re-routed to available paths during a path failure
- Ensures that the failed paths are revalidated as soon as possible
- Configures the multipaths to maximize performance
- Reconfigures the multipaths automatically when events occur
- Provides DMMP features support to newly added logical unit
- Provides device name persistency for DMMP devices under `/dev/mapper/`
- Configures multipaths automatically at an early stage of rebooting to permit the OS to install and reboot on a multipathed logical unit

Known Limitations and Issues of the Device Mapper

- When storage is configured with AVT mode, delays in device discovery might occur. Delays in device discovery might result in long delays when the operating system boots.
- In certain error conditions with `no_path_retry` or `queue_if_no_path` feature set, applications might hang forever. To overcome these conditions, you must enter the following command to all the affected multipath devices: `dmsetup message device 0 "fail_if_no_path"`, where `device` is the multipath device name (for example, mpath2; do not specify the path).
- An I/O hang might occur when a volume is unmapped without first deleting the DM device.

NOTE This limitation applies to only the SUSE 11 OS.

- Stale entries might not be noticed in multipath `-ll` output if the volumes are unmapped or deleted without first deleting the DM device and its underlying paths.

NOTE This limitation applies to only the SUSE 11 OS.

- Currently, the `mode select` command is issued synchronously for each LUN. With large LUN configurations, slower failovers for DM multipath devices might occur if there is any delay in completing of the `mode select` command.

NOTE This limitation applies to only the SUSE 11 OS.

- If the scsi_dh_rdac module is not included in initrd, slower device discovery might occur, and the syslog might get populated with buffer I/O error messages.
- If the storage vendor and model are not included in scsi_dh_rdac device handler, slower device discovery might be seen, and the syslog might get populated with buffer I/O error messages.
- Use of the DMMP and RDAC failover solutions together on the same host is not supported. Use only one solution at a time.

Installing the Device Mapper Multi-Path

1. Use the media supplied by your operating system vendor to install SLES 11.
2. Install the errata kernel 2.6.27.29-0.1.

   Refer to the *SUSE Linux Enterprise Server 11 Installation and Administration Guide* for the installation procedure.
3. To boot up to 2.6.27.29-0.1 kernel, reboot your system.
4. On the command line, enter `rpm -qa |grep device-mapper`, and check the system output to see if the correct level of the device mapper component is installed.
   - **The correct level of the device mapper component is installed** – Go to step 5.
   - **The correct level of the device mapper component is not installed** – Install the correct level of the device mapper component or update the existing component, and go to step 5.
5. On the command line, enter `rpm -qa |grep multipath-tools`, and check the system output to see if the correct level of the multipath tools is installed.
   - The correct level of the multipath tools is installed – Go to step 6.
   - The correct level of the multipath tools is not installed – Install the correct level of the multipath tools or update the existing multipath tools, and go to step 6.
6. Update the configuration file `/etc/multipath.conf`.

   See Setting Up the multipath.conf File on page 46 for detailed information about the `/etc/multipath.conf` file.
7. On the command line, enter `chkconfig multipathd on`.

   This command enables multipathd daemon when the system boots.
8. Edit the `/etc/sysconfig/kernel` file to add `directive scsi_dh_rdac` to the INITRD_MODULES section of the file.
9. Download the KMP package for scsi_dh_rdac for the SLES 11 architecture from the website http://forgeftp.novell.com/driver-process/staging/pub/update/lsi/sle11/common/, and install the package on the host.
10. Update the boot loader to point to the new initrd image, and reboot the host with the new initrd image.

Setting Up the multipath.conf File

The multipath.conf file is the configuration file for the multipath daemon, multipathd. The multipath.conf file overwrites the built-in configuration table for multipathd. Any line in the file whose first non-white-space character is # is considered a comment line. Empty lines are ignored.

Installing the Device Mapper Multi-Path for SLES 11.1

All of the components required for DMMP are included in SUSE Linux Enterprise Server (SLES) version 11.1 installation media. However, users might need to select the specific component based on the storage hardware type. By default, DMMP is disabled in SLES. You must follow the following steps to enable DMMP components on the host.

1. On the command line, type `chkconfig multipath on`.

   The multipathd daemon is enabled with the system starts again.
2. Edit the /etc/sysconfig/kernel file to add the directive scsi_dh_rdac to the INITRD_MODULES section of the file.
3. Create a new initrd image using the following command to include scsi_dh_rdac into ram disk:

   ```
   mkinitrd -i /boot/initrd-r -rdac -k /bootvmlinuz
   ```
4. Update the boot leader to point to the new initrd image, and reboot the host with the new initrd image.

Copy and Rename the Sample File

Copy and rename the sample file located at /usr/share/doc/packages/multipath-tools/multipath.conf.synthetic to /etc/multipath.conf. Configuration changes are now accomplished by editing the new /etc/multipath.conf file. All entries for multipath devices are commented out initially. The configuration file is divided into five sections:

- **defaults** – Specifies all default values.
- **blacklist** – All devices are blacklisted for new installations. The default blacklist is listed in the commented-out section of the /etc/multipath.conf file. Blacklist the device mapper multipath by WWID if you do not want to use this functionality.
- **blacklist_exceptions** – Specifies any exceptions to the items specified in the section blacklist
- **devices** – Lists all multipath devices with their matching vendor and product values
- **multipaths** – Lists the multipath device with their matching WWID values

Determine the Attributes of a MultiPath Device

To determine the attributes of a multipath device, check the multipaths section of the /etc/multipath.conf file, then the devices section, then the defaults section. The model settings used for multipath devices are listed for each storage array and include matching vendor and product values. Add matching storage vendor and product values for each type of volume used in your storage array.

For each UTM LUN mapped to the host, include an entry in the blacklist section of the /etc/multipath.conf file. The entries should follow the pattern of the following example.

```
blacklist {
device {
        vendor "*"
        product "Universal Xport"
   }
}
```

The following example shows the devices section for LSI storage from the sample `/etc/multipath.conf` file. Update the vendor ID, which is LSI in the sample file, and the product ID, which is `INF-01-00` in the sample file, to match the equipment in the storage array.

```
devices {
    device {
        vendor                      "LSI"
        product                     "INF-01-00"
        path_grouping_policy        group_by_prio
        prio                        rdac
        getuid_callout            "/lib/udev/scsi_id -g -u -d /dev/%n"
        polling_interval            5
        path_checker                rdac
        path_selector               "round-robin 0"
        hardware_handler            "1 rdac"
        failback                     immediate
        features                    "2 pg_init_retries 50"
        no_path_retry                30
        rr_min_io                    100
    }
}
```

The following table explains the attributes and values in the devices section of the `/etc/multipath.conf` file.

Table 15 Attributes and Values in the multipath.conf File

Attribute	Parameter Value	Description
`path_grouping_policy`	`group_by_prio`	The path grouping policy to be applied to this specific vendor and product storage.
`prio`	`rdac`	The program and arguments to determine the path priority routine. The specified routine should return a numeric value specifying the relative priority of this path. Higher numbers have a higher priority.
`getuid_callout`	`"/lib/udev/ scsi_id -g -u -d /dev/%n"`	The program and arguments to call out to obtain a unique path identifier.
`polling_interval`	`5`	The interval between two path checks, in seconds.
`path_checker`	`rdac`	The method used to determine the state of the path.
`path_selector`	`"round-robin 0"`	The path selector algorithm to use when there is more than one path in a path group.
`hardware_handler`	`"1 rdac"`	The hardware handler to use for handling device-specific knowledge.
`failback`	`10`	A parameter to tell the daemon how to manage path group failback. In this example, the parameter is set to 10 seconds, so failback occurs 10 seconds after a device comes online. To disable the failback, set this parameter to `manual`. Set it to `immediate` to force failback to occur immediately.

Attribute	Parameter Value	Description
`features`	`"2 pg_init_retries 50"`	Features to be enabled. This parameter sets the kernel parameter `pg_init_retries` to `50`. The `pg_init_retries` parameter is used to retry the mode select commands.
`no_path_retry`	`30`	Specify the number of retries before queuing is disabled. Set this parameter to `fail` for immediate failure (no queuing). When this parameter is set to `queue`, queuing continues indefinitely.
`rr_min_io`	`100`	The number of I/Os to route to a path before switching to the next path in the same path group. This setting applies if there is more than one path in a path group.

Using the Device Mapper Devices

Multipath devices are created under `/dev/` directory with the prefix `dm-`. These devices are the same as any other bock devices on the host. To list all of the multipath devices, run the `multipath –ll` command. The following example shows system output from the `multipath –ll` command for one of the multipath devices.

```
mpathp (3600a0b80005ab177000017544a8d6b92) dm-0 LSI,INF-01-00
[size=5.0G][features=3 queue_if_no_path
pg_init_retries 50][hwhandler=1 rdac][rw]
\_ round-robin 0 [prio=6][active] \_ 5:0:0:0
sdc  8:32   [active][ready] \_
round-robin 0 [prio=1][enabled] \_ 4:0:0:0   sdb  8:16
[active][ghost]
```

In this example, the multipath device node for this device is `/dev/mapper/mpathp` and `/dev/dm-0`. The following table lists some basic options and parameters for the multipath command.

Table 16 Options and Parameters for the `multipath` Comand

Command	Description
`multipath -h`	Prints usage information
`multipath -ll`	Shows the current multipath topology from all available information (sysfs, the device mapper, path checkers, and so on)
`multipath -f map`	Flushes the multipath device map specified by the map option, if the map is unused
`multipath -F`	Flushes all unused multipath device maps

Troubleshooting the Device Mapper

Table 17 Troubleshooting the Device Mapper

Situation	Resolution
Is the multipath daemon, multipathd, running?	At the command prompt, enter the command: `/etc/init.d/multipathd status`.
Why are no devices listed when you run the `multipath -ll` command?	At the command prompt, enter the command: `#cat /proc/scsi/scsi`. The system output displays all of the devices that are already discovered. Verify that the `multipath.conf` file has been updated with proper settings.

Chapter 10: Failover Drivers for the Solaris Operating System

MPxIO is the supported failover driver for the Solaris operating system.

Solaris OS Restrictions

SANtricity ES Storage Manager no longer supports or includes RDAC for these Solaris operating systems:

- Solaris 10 OS
- Solaris 9 OS
- Solaris 8 OS

NOTE MPxIO is not included on the SANtricity ES Storage Manager Installation DVD.

Prerequisites for Installing MPxIO on the Solaris OS for the First Time

Perform these prerequisite tasks.

1. Install the hardware
2. Map the volumes.
3. Make sure that RDAC is not on the system, because you cannot run both RDAC and MPxIO.

NOTE RDAC and MPxIO cannot run on the same system.

Prerequisites for Installing MPxIO on a Solaris OS That Previously Ran RDAC

Perform these prerequisite tasks:

1. Make sure that there are no major problems in the current RDAC.

ATTENTION Potential loss of data access – Some activities, such as adding and removing storage arrays, can lead to stale information in the RDAC module name file. These activities might also render data temporarily inaccessible.

2. To make sure that there are no leftover RDAC files, type this command, and press Enter:
   ```
   ls -l /var/symsm/directory
   ```
3. To make sure that the RDAC directory does not exist, type this command, and press Enter:
   ```
   ls -l /dev/rdsk/*s2 >>filename
   ```
4. Examine the `/etc/symsm/mnf` file. There should be one line for each currently connected storage array. An example line is:
   ```
   infiniti23/24~1T01610104~ 0 1 7~1T04110240~ 7~0~3~~c6t3d0~c4t2d7~
   ```
5. Make sure that there are no extra lines for disconnected storage arrays.
6. Make sure that two controllers are listed on each line. (The example shows c6t3 and c4t2.)
7. Make sure that these are the correct controllers.
8. Make sure there are no identical cXtX combinations. For example, if you see c6t3d0 and c6t3d4, a problem exists.

9. Make sure that no major problems exist in the Solaris OS.

   When you reset or power-cycle a controller, it might take up to three minutes to come fully ready. Older storage arrays might take longer. The default Solaris Not Ready retry timer is only 30 seconds long, so spurious controller failovers might result. Sun Microsystems has already increased the timer for LSI-branded storage arrays to two minutes.
10. Make sure that the VERITAS Volume Manager can handle the MPxIO upgrade.
11. Capture the current configuration by performing these steps:
    a. Save this file.

       `/etc/symsm/mnf`
    b. Save this file.

       `/etc/raid/rdac_address`
    c. Type this command, and press Enter:

    ```
    ls -l /dev/rdsk/*s2 >>rdsk.save
    ```
    d. Type this command, and press Enter:

    ```
    ls -l /dev/symsm/dev/rdsk/*s2 >>symsm.save
    ```
    e. Type this command, and press Enter:

    ```
    lad -y >>lad.save
    ```
12. To remove RDAC, type this command, and press Enter:

    ```
    pkgrm RDAC
    ```

NOTE Do not restart the system at this time.

Installing MPxIO on the Solaris 9 OS

MPxIO is not included in the Solaris 9 OS. To install MPxIO on the Solaris 9 OS, perform these steps.

1. Download and install the *SAN 4.4x release Software/Firmware Upgrades and Documentation* from this website:

   `http://www.sun.com/storage/san/`
2. Install recommended patches.
3. To enable MPxIO on the Solaris 9 OS, perform these steps:
   a. Open this file:

      `/kernel/drv/scsi_vhci.conf`
   b. Change the last line in the script to this command:

   ```
   mpxio-disable="no";
   ```

NOTE Make sure that `"no"` is enclosed in double quotation marks.

4. Disable MPxIO on any Fibre Channel drives, such as internal drives, that should not be MPxIO enabled. To disable MPxIO on specific drives, perform these steps:
   a. Open the Fibre Channel port driver configuration file:

      `/kernel/drv/qlc.conf`
   b. Add a line similar to the following:

   ```
   name="qlc" parent="/pci@8,600000" port=0 unit-address="2"
   mpxio-disable="yes";
   ```

NOTE To find the correct parent and port numbers, look at the device entry for the internal drives, found in `/dev/dsk`,

5. To update vfstab and the dump configuration, type this command:

```
stmsboot -u
```

6. Reboot the system.

Enabling MPxIO on the Solaris 10 OS

MPxIO is included in the Solaris 10 OS; therefore, it does not need to be installed. It only needs to be enabled.

1. To enable MPxIO for all Fibre Channel drives, enter the command:

```
stmsboot -e
```

2. If there are any Fibre Channel drives for which you do not want MPxIO enabled, for example, internal drives, disable MPxIO on those drives. To disable MPxIO on specific drives, perform these steps:
   a. Edit the Fibre Channel port driver configuration file `/kernel/drv/fp.conf`.
   b. Add a line similar to the following:

```
name="fp" parent="/pci@8,600000/SUNW,qlc@2" port=0
mpxio-disable="yes";
```

   To find the correct parent and port numbers, look at the device entry for the internal drives, found in `/dev/dsk`.

3. To update vfstab and the dump configuration, enter the command:

```
stmsboot -u
```

4. Reboot the system.

Configuring Failover Drivers for the Solaris OS

ATTENTION **Possible loss of data** – Create a backup of your configuration before you change any configuration file.

Configure the host settings for these parameters in the `/etc/` system file.

- `ssd_io_time` – Add the line `set ssd:ssd_io_time=0x78` to the system file.
- `ssd_max_throttle` – Add the line `set ssd:ssd_max_throttle=8` to the system file.

Configure MPxIO multi-pathing in the `/kernel/drv/scsi_vhci.conf` file with these recommended parameters.

- `mpxio-disable` – Set to no.
- `load-balance` – Set to none.

NOTE For a symmetric storage array, you must specify round-robin so that the driver can balance the I/O load between the two paths.

- `auto-failback` – Set to `enable`.

To add a device, add these lines to the end of the `scsi_vhci.conf` file:

```
device-type-scsi-options-list="Acme     MSU",
"symmetric-option";symmetric-option=0x1000000;
```

where Acme is the vendor ID, and MSU is the product ID.

NOTE Make sure five spaces exist between the vendor ID and the product ID.

Frequently Asked Questions About Solaris Failover Drivers

Table 18 Frequently Asked Questions about Solaris Failover Drivers

Question	Answer

Where can I find MPxIO-related files?	You can find MPxIO-related files in these directories:
	`/etc/`
	`/kernel/drv`

Where can I find data files?	You can find data files in these directories:
	`/etc/raid ==> /usr/lib/symsm`
	`/var/symsm`
	`/var/opt/SM`

Where can I find the command line interface (CLI) and bin files?	You can find CLI and bin files in these directories:
	`etc/raid/bin ==> /usr/lib/symsm/bin`
	`/usr/sbin/symsm ==> /usr/lib/symsm/bin`

Where can I find device files?	You can find device files in this directory: `/dev/[osa|symsm]/dev/[r]dsk ==>/dev/[r]dsk`

Where can I find the SANtricity ES Storage Manager files?	You can find the SANtricity ES Storage Manager files in these directories:
	`/opt/SM7[client, agent]`
	`/opt/SM9`

How can I get a list of controllers and their volumes?	Use the `lad -y` command. The command uses LUN 0 to get the information and is located in the `/usr/lib/symsm/bin` directory. It can be reached through `/etc/raid/bin`. This command updates the `mnf` file.
Where can I get a list of storage arrays, their volumes, LUNs, WWPNs, preferred paths, and owning controller?	Use the `SMdevices` utility. This utility must be in the search path and is located in the `/opt/SM7util` directory or the `/opt/SM8/util` directory.
How can I see if volumes have been added?	Use the `hot_add` utility. This utility asks the Solaris OS target drivers to find new devices. The `hot_add` utility works only after all of the potential devnodes have been created in the OS. If not, run the `devfsadm` command, and run the hot_add utility. If these actions do not work, restart with the reconfiguration option.

Question	Answer
What file holds the storage array identification information?	Go to the `/etc/[osa|symsm]/mnf` directory. The `mnf file` identifies storage arrays in these ways: ■ Lists their ASCII names ■ Shows their controller serial numbers ■ Indicates the current LUN distribution ■ Lists controller system names ■ Lists the storage array numbers
Why might the rdriver fail to attach and what can I do about it?	The rdriver might not attach if there is no entry in the `rdriver.conf` file to match the device, or if rdnexus runs out of buses. If no physical devnode exists: ■ The `sd.conf` file must specify LUNs explicitly. ■ The `ssd.conf` file with the itmpt HBA must specify LUNs explicitly. ■ With Sun driver stacks, underlying HBA drivers dynamically create devnodes for ssd. You must restart the system with the reconfigure option after you update the `rdriver.conf` file. In the Solaris 9 OS, you can use the `update_drv` command if the HBA driver supports it.
How can I determine if the resolution daemon is working?	Type this command, and press Enter: `ps -ef | grep rd`

Printed in U.S.A.